WIRELESS COMMUNICATION PROTOCOL ARCHITECTURES FOR NANOSENSOR NETWORKS

By

Zhongping Zhang

RECOMMENDED:

____

____

____

Advisory Committee Chair

____

Chair, Department of Electrical and Computer Engineering

APPROVED:

____

Dean, College of Science, Engineering and Mathematics

____

Dean of the Graduate School

May 3, 2004

Date

WIRELESS COMMUNICATION PROTOCOL ARCHITECTURES FOR NANOSENSOR NETWORKS

A

THESIS

Presented to the Faculty

of the University of Alaska Fairbanks

in Partial Fulfillment of the requirements

for the Degree of

MASTER OF SCIENCE

By

Zhongping Zhang, B.S.

Fairbanks, Alaska

May 2004

ABSTRACT

Recent developments in micro fabrication and nanotechnology will enable the inexpensive manufacturing of massive numbers of tiny computing elements with sensors. New programming paradigms are required to obtain organized and coherent behavior from the cooperation of large numbers of sensor nodes. The individual nodes are identical, randomly placed and unreliable. They communicate with a small local neighborhood via wireless broadcast. In such environments, where individual nodes have limited resources, aggregating the node into groups is useful for specialization, increased robustness, and efficient resource allocation. In this paper, an application-specific self-organization protocol stack is developed.

The clustering process is divided into phases. The first phase is to know the neighbor nodes. The second phase is to set up the cluster and routing. A "find maximum clique algorithm" is used to set up clusters. A back off method is used to set up the hop field and routing. Group leaders set up a TDMA schedule for steady state operation. This schedule ensures that there is no conflict among in the same cluster and between clusters. Direct-sequence spread spectrum (DS-SS) is used to avoid inter-group conflict.

The limited power resource is a challenge in nanosensor network. This paper use two different ways to analyze energy consumed in nanosensor networks, energy cost field and bit flow method. Sensor node deployment, cluster size, and propagation condition effect are discussed in this paper by those two methods respectively.

TABLE OF CONTENTS

Signature	
Title	
Abstract	
Table of contents	
List of Figures	viii
List of Tables	x
List of Appendices	x
Acknowledgements	xi
I. Introduction	1
A. Possible applications of nanosensor networks	1
1. Military applications	2
2. Home applications	2
3. Health applications	2
4. Environment applications	3
B. Microsensor network characteristics and challenges	3
C. Sensor node framework and sensor network communication model	4
1. Sensor node framework/architecture	4
2. Sensor network communication model	5

D.Microsensor networks	7
II. Background	11
A. Radio wave propagation	11
1. Free space model	12
2. Two-ray model	13
3. The shadowing model	15
B. Link-layer Protocols	17
1. Block codes	18
2. Convolutional codes	19
C. Media Access control (MAC) Protocols	19
1. FDMA	20
2. TDMA	21
3. CDMA	23
4. Pure Aloha	26
5. Slotted Aloha	26
6. CSMA	27
7. MACA	28
8. MACAW	28
D. Routing protocols	30
1. Introduction	30
2. Destination Sequenced Distance Vector	31
3. Global State Routing	32

4. Dynamic Source Routing 32

5. The Ad-Hoc On Demand Distance Vector 33

III. The Protocol Stack 34

A. Assumptions 34

B. Overview 34

C. Clustering Protocols 36

1. Phase I -- to know neighbors 37

2. Phase II -- set up cluster and routing 39

D. Steady State Protocols 50

1. Intra-cluster communication 50

2. Inter-cluster communication 50

E. Features and properties of protocol stack 56

IV. Analysis 58

A. Sensor node deployment 58

B. Optimum cluster size 66

1. Radio link budget 66

2. Transport bit energy 70

3. Optimization of cluster size 71

C. Propagation condition 76

D. Data compression ratio 81

1. Data correction 82

2. Data aggregation 84

V. Conclusions and Results 86

References 87

Appendices 90

LIST OF FIGURES

Figure 1-1. Sensor node framework	5
Figure 1-2. A sensor network scene	6
Figure 1-3. Multihop infrastructureless architecture	6
Figure 1-4. Hierarchical clustering	7
Figure 2-1. Two-ray model over flat earth	13
Figure 2-2. FDMA scheme	21
Figure 2-3. TDMA scheme	22
Figure 2-4. CDMA scheme	24
Figure 2-5. Near-far effect illustrated	25
Figure 2-6. Pure Aloha Protocol	26
Figure 2-7. Slotted Aloha Protocol	27
Figure 3-1. Cluster-based multihop scene	36
Figure 3-2. Two kind of communication slots	36
Figure 3-3. State chart of knowing neighbors	37
Figure 3-4. A 9-node simple sensor network	38
Figure 3-5. A simple maximum clique algorithm	42
Figure 3-6. The neighbors of node 6	43
Figure 3-7. Set up cluster and routing	49
Figure 3-8. Intra-cluster slots	51
Figure 3-9. Transmission power sensitivity	53

Figure 3-10. Inter-cluster slots 54

Figure 4-1. Simulation of uniform sensor nodes (150) deployment 59

Figure 4-2. Calculation of the energy cost field 61

Figure 4-3. Energy cost density graphic 64

Figure 4-4. 3-D energy cost density graphic 65

Figure 4-5. Simulation of optimization sensor nodes (150) deployment 66

Figure 4-6. Calculation of the bit energy cost 73

Figure 4-7. Hop density function 74

Figure 4-8. Energy cost vs. cluster size ($P_r \propto 1/d^2$) 75

Figure 4-9. Propagation condition effect in bit energy cost (1) 78

Figure 4-10. Propagation condition effect in bit energy cost (2) 79

Figure 4-11. Propagation condition effect in bit energy cost (3) 80

Figure 4-12. Propagation condition effect in bit energy cost (4) 81

Figure 4-13. Bit energy cost vs. compression ratio 82

Figure 4-14. Minimum data correlation 84

LIST OF TABLES

Table 2-1. Some typical values of path loss exponent, σ_{dB} 17

Table 2-2. Some typical values of shadowing deviation 17

Table 3-1. The first phase communication of the 9-node simple sensor network 39

LIST OF APPENDIES

Appendix 1. Uniform deployment of sensor nodes 90

Appendix 2. Energy cost field function 91

Appendix 3. Optimum deployment of sensor nodes 92

Appendix 4. Bit energy cost while inter-clust pass loss Exponent d = -1.5 to –4 93

Appendix 5. Bit energy cost while inter-clust pass loss Exponent d = -5 to –7 94

ACKNOWLEDGEMENTS

I would like to express my gratitude for the efforts and dedication from my advisor, Dr. Chales E. Mayer. Dr. Mayer has provided me with support and was an endless source of ideas. I think him for the countless hours he has spent with me, discussing my research, reading my paper. His assistance has been invaluable.

I would also like to thank Dr. William Bristow and Dr. Denise Thorsen, my co-advisors. Their enthusiasm for research has been an inspiration. I would thank them for their support and help.

I would like to thank the Center for Nanosensor Technology for their support.

Finally there are many thanks that cannot be expressed in words to my parents and wife for all their love they have had in me.

I Introduction

A. Possible applications of nanosensor networks

Recent advances in micro-electro-mechanical systems (MEMS) technology, wireless communications, and digital electronics have enabled the development of low-cost, low-power, multifunctional sensor nodes that are small in size and communicate untethered over short distances. Sensor nodes can be randomly dispersed over an area of interest and are capable of RF communication. They can contain signal processing engines to manage the communication protocols and for data processing before transmission. The individual nodes have a limited capability, but are capable of achieving much larger functionality through coordinated effort in a network that typically consists of hundreds to thousands of nodes. Sensor networks may consist of many different types of sensors such as magnetic, thermal, visual, acoustic, infrared, radar and seismic. Sensor networks are able to monitor a wide variety of ambient conditions, such as speed, mechanical stress levels, noise levels, soil makeup, pressure, lightning conditions, temperature and humidity. Sensor networks can be used in a wide range of application domains, because of their reliability, accuracy, flexibility, cost-effectiveness and ease of deployment. According to Tilak et al. [1], microsensor networks can detect and collect data concerning any signs of machine failure, earthquake, flood, and even a terrorist attack. Sensor networks enable: (1) information gathering; (2) information processing; (3) and reliable monitoring of a variety of environments for both civil and military applications. Those applications can be categorized into military, home, health, environment, disaster relief, space exploration and chemical processing [2].

1. Military applications

Since sensor networks are based on the dense deployment of disposable and low-cost sensor nodes, destruction of some nodes by hostile actions does not affect a military operation as much as the destruction of a traditional sensor. Sensor networks thus afford a good approach for battlefields. Wireless sensor networks can be an integral part of military command, control, communications, computing, intelligence, surveillance, reconnaissance and targeting (C4ISRT) systems. Sensor networks can be used as a chemical or biological warning or reconnaissance system. Sensor networks can be deployed in critical terrains, and some valuable, detailed, and timely intelligence about the opposing force and terrain can be gathered within minutes.

2. Home applications

Domestic devices, such as microwave ovens, refrigerators and vacuum cleaners, can be embedded with some smart sensor nodes, which can interact with each other and with the external network via the internet or satellite. They allow end users to manage home devices locally and remotely more easily. The sensor nodes and room servers can be integrated with existing embedded devices to become self-organizing, self-regulated, and adaptive systems based on control theory models.

3. Health applications

Sensor networks can collect and store physiological data. Those data can be used for medical exploration. The installed sensor networks can also monitor and detect a patient's behavior. These small sensor nodes can also help doctors identify pre-defined symptoms earlier.

4. Environment applications

Sensor networks can relay the exact origin of a fire to the end users before the fire is spread uncontrollably. Those sensor nodes may be equipped with an effective power scavenging device, such as solar cells so that they can maintain for months and even years. Sensor networks can also be used to help get integrated information across temporal and spatial scales to form a biocomplexity mapping of the environment. Rainfall, water level and weather sensors can be used to detect a flood.

B. Microsensor network characteristics and challenge.

Most sensor networks encounter the following operational characteristics and challenges [3]:

- Consists of hundreds to thousands of densely deployed sensors that are designed for unattended operation. The problem can be viewed in terms of collision and congestion.
- Sensors are prone to failure.
- Sensors may be stationary or mobile after deployment. Sensor networks' topology may change very frequently.
- The traffic is of a statistical nature; the data rate is expected to be very low.

- Sensors carry limited, generally irreplaceable power resources. Computational capacities and memory are also limited.
- The position of sensors may be pre-determined or randomly deployed. Ad hoc deployment requires that the system identify and copes with the resulting distribution and connectivity of nodes.
- Sensors use a broadcast communication paradigm.
- Dynamic environmental conditions require the system to adapt over time to changing connectivity and system stimuli.

C. Sensor framework and sensor network communication model

1. Sensor node framework/architecture

A sensor node is made up of four basic components: a sensing unit, a processing unit, a transceiver unit and a power unit. They may also have application dependent additional components such as a location finding system, a power generator and a mobilizer, as shown in figure 1-1. Sensing units are usually composed of two subunits: sensors and analog to digital converters (ADCs). The processing unit, which is generally associated with a small storage unit, manages the procedures that make the sensor node collaborate with the other nodes to carry out the assigned sensing tasks [2].

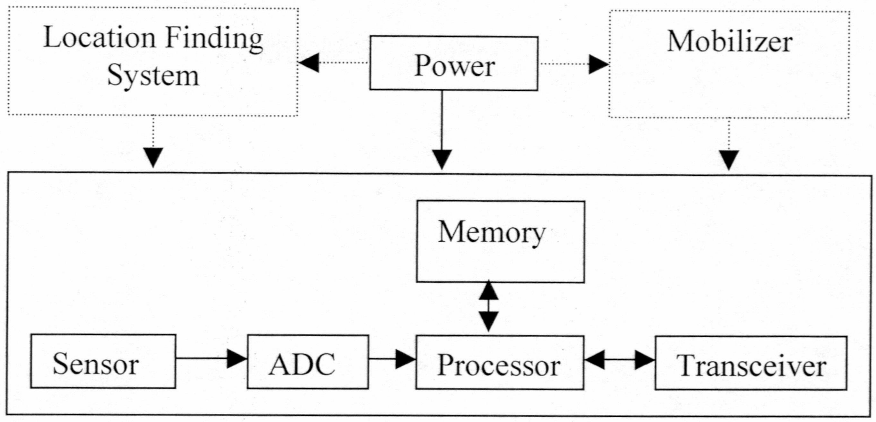

Figure 1-1. Sensor node framework.

Analog data which is obtained from the sensor is converted to digital by an ADC before it enters into the processor. Because the energy expenditure in data processing is much less compared to data communication, only processed, necessary data are transmitted through a transceiver to communicate with other nodes. Power components are usually irreplaceable batteries, and may be supported by power scavenging components, such as solar cells, vibration to electricity converters, or piezo-electrical components. When knowledge of location is required, a location finding system is necessary. A mobilizer can be used to move the sensor node.

2. Sensor network communication model

The sensor nodes are usually scattered in a sensor field as shown in figure 1-2. Each of these sensor nodes can collect data and route data back to the sink and the end user [2], [3].

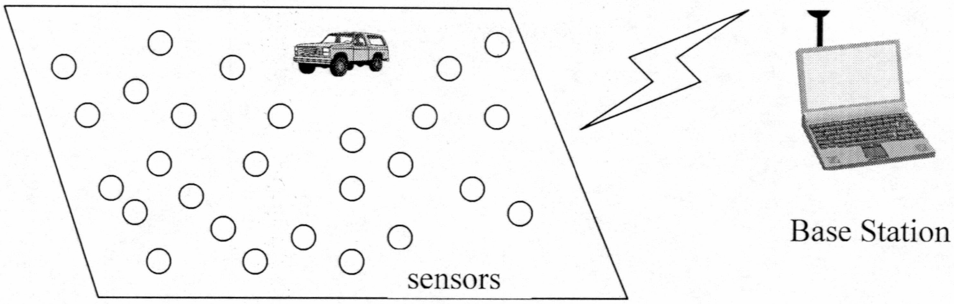

Figure 1-2. A sensor network scene.

In some small sensor networks, each sensor communicates with the base station directly. Sensor nodes do not cooperate with each other. In most situations, sensor nodes are highly constrained in resources. They cooperate in the transfer of sensed data or processed information to the base station.

One kind of this cooperation is multihop. When a sensor node has data to send to a base station, it transmits its data to neighbors, then its neighbor forwards the data further. For example, in figure 1-3, node A sends its data to base station through node B, C, D and E.

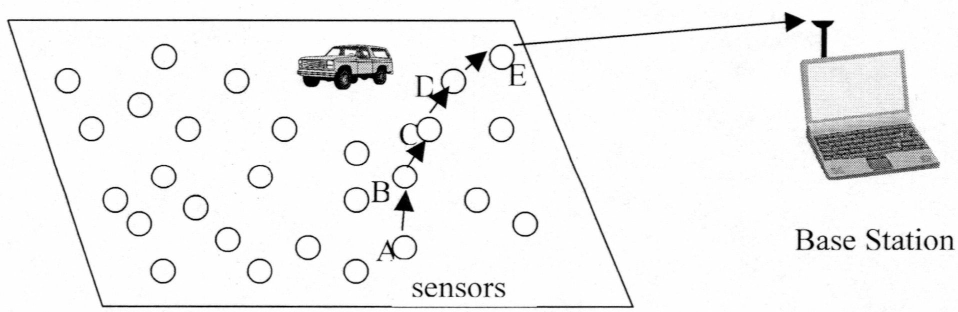

Figure 1-3. Multihop infrastructureless architecture.

In some sensor networks, sensor nodes autonomously form groups called clusters. This clustering process is applied recursively to form a hierarchy of clusters. This hierarchical structure is shown in figure 1-4. The sensor network is divided into clusters. Cluster heads are grouped into the second level cluster and so on. The base station is at the top level [3].

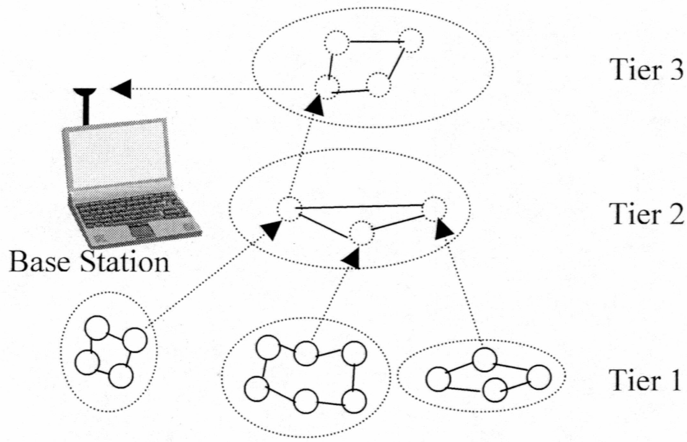

Figure 1-4. Hierarchical clustering.

D. Microsensor Networks

Scalable Coordination Architectures for Deeply Distributed Systems (SCADDS) [4] is a research project at the Information Sciences Institute, University of Southern California (USC/ISI). The SCADDS project explores scalable coordination architectures for deeply distributed and dynamic systems. These systems have heterogeneous nodes, which have a range of sensing, actuation and communication capabilities. A novel data-centric, data dissemination paradigm for sensor networks, called directed diffusion, was developed. Directed diffusion consists of several elements. Data is named using attribute-value pairs. A sensing task (or a subtask thereof) is disseminated throughout the sensor network as an interest for named data. This dissemination sets up gradients within the network designed to "draw" events (i.e., data matching the interest). Events start flowing towards the originators of interests along multiple paths. The sensor network reinforces one, or a small number of these paths.

μ-Adaptive Multi-domain Power aware Sensors (μ-AMPS) [5] is a research project at Massachusetts Institute of Technology (MIT). This project comes up with techniques to minimize power consumption at the three levels: Node architecture level, Network level and software level. The project uses a sensor node that consists of a battery with DC-DC conversion to the appropriate voltages required by the system. The power-aware system is sentient of the many variables that define the energy consumption at each architectural block, from leakage currents in the integrated circuits, to the output quality and latency requirements of the end user, to the duty cycles of radio transmission.

PicoRadio [6] is a project at Berkeley Wireless Research Center, University of California, Berkeley. PicoRadio project is focused on developing meso-scale low cost (< 50 cents) transceivers for ubiquitous wireless data acquisition that minimizes energy and power dissipation. This project comes up with innovations at the protocol stack level which make the intended energy reductions possible.

Smart Dust [7] is a project at Electrical Engineering and Computer Sciences, University of California, Berkeley (UCB/EECS). The goal of the Smart Dust project is to build a self-contained, millimeter-scale sensing and communication platform for a massively distributed sensor network. Smart Dust uses free-space optical communications for wireless networking. Integrated into a dust mote are MEMS sensors, a semiconductor laser diode and MEMS beam-steering mirror for active optical transmission, a MEMS corner-cube retroreflector for passive optical transmission, and optical receiver, signal-processing and control circuitry, and a power source based on thick-film batteries and solar cells. This remarkable package has the ability to sense and communicate. Smart Dust uses optical reflector technology to transmit passively. This provides an inexpensive way to probe a sensor or acknowledge that information was received.

Wireless Integrated Networks Sensors (WINS) [8] project is at UCLA and is sponsored by DAPRPA and the Air Force Research Laboratory. WINS provides distributed network and Internet access to sensors, controls, and processors that are deeply embedded in equipment, facilities, and the environment. WINS combine microsensor technology, low power signal processing, low power computation, and low power, low cost wireless networking capability in a compact system. WINS networks will provide sensing, local control, and embedded intelligent systems in structures, materials, and environments. WINS nodes are distributed at high density in an environment to be monitored. Multi-hop communication permits low power operation of dense WINS sensor networks. The

WINS sensor systems operate at low power, sampling at low frequency, and with environmental background limited sensitivity. The work done in the AWAIRS project was used in the WINS project. Low-Energy Adaptive Clustering Hierarchy (LEACH) protocol is developed for this project. LEACH is a protocol architecture for wireless microsensor networks that achieves low energy dissipation and latency without sacrificing application-specific quality. LEACH uses a randomized, adaptive, self configuring algorithm to format clusters. Once clusters have been formed, the nodes must communicate their data to the cluster head node using a time-division multiple access (TDMA) protocol, which allows the nodes to shut down some internal components and enter a sleep state when they are not transmitting data to the cluster head. In addition, using a TDMA approach in the steady state ensures there are no collisions of data within the cluster, saving energy and time. The cluster head is responsible for receiving all the data from nodes within the cluster and aggregating this data into a smaller set of information that describes the events the nodes are sensing. Thus the cluster head node takes a number of data signals and reduces the actual data (total number of bits) while maintaining the effective data (information content). The cluster head node must then send the aggregate data set to the destination.

II. Background

A. Radio wave propagation

Unlike wired channels that are predictable, radio channels are extremely random. The transmission path between the transmitter and the receiver can vary from simple line-of-sight to one that is severely obstructed by buildings, mountains, and foliage.

The mechanisms behind electromagnetic wave propagation can generally be attributed to reflection, diffraction, and scattering. Reflection occurs when waves impinge upon an object that has a large dimension compared to its own wavelength. Reflection occurs from the surface of the earth, buildings, and walls. Diffraction occurs when the wave propagation path is obstructed by a sharp edge. This gives rise to bending of waves around the obstacles even when there is no line-of-sight present. Diffraction depends on the wavelength, phase, amplitude, and polarization of the incident waves. Scattering occurs when objects with dimensions small compared with the wavelength get in the way of the propagation path, and when the number of obstacles per unit volume is large. In general these objects include foliage, street lamps, and small particles such as dust, etc. [9], [10]. Modeling the radio channel has been one of the most difficult tasks. Most of modeling is done in a statistical fashion based on measurements.

Propagation models have traditionally focused on predicting the average received signal strength at a given distance from the transmitter, as well as the variability of the signal strength in close spatial proximity to a particular location.

1. Free space model

The most basic model of radio wave propagation involves so called "free space" radio wave propagation [11]. In this model, radio waves emanate from a point source of radio energy, traveling in all directions in a straight line, filling the entire spherical volume of space with radio energy that varies in strength. The free space model predicts an inverse distance squared relationship as the received power $P_r(d)$ decays as a function of the transmitter-receiver separation distance d as

$$P_r(d) = \frac{P_t G_t G_r}{\left(\frac{4\pi d}{\lambda}\right)^2} \qquad \text{Eqn. 2-1}$$

where,

$P_r(d)$ is the receive power given a transmitter-receiver separation of d,

P_t is the transmit power,

G_t is the gain of the transmitting antenna,

G_r is the gain of receiving antenna,

λ is the wavelength of the carrier signal, and

d is the distance between the transmitter and the receiver.

The free space propagation model assumes the ideal propagation condition that there is only one clear unobstructed line-of-sight path between the transmitter and receiver.

2. Two-Ray model

A single line-of-sight path between two mobile nodes is seldom the only means of propagation. The two-ray ground reflection model can be used to represent transmission where there is a single direct path between a transmitter and receiver and a single strong reflected path, often from the ground, as shown in figure 2-1. This model is based on some assumptions, such as the antenna is not too high, the combined horizon distance between transmitter and receiver is not small (few kilometers), the earth surface is smooth and flat. The typical conditions for two-ray model are : antenna height equals about 3 meter, distance is about 2 kilometers, and frequency is 900 MHz.

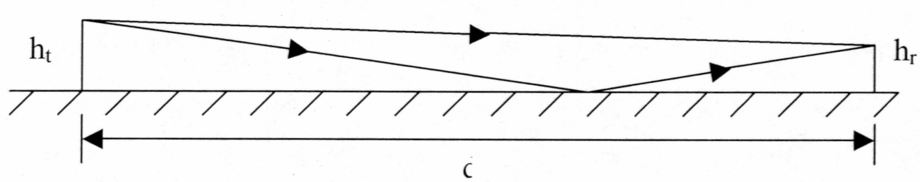

Figure 2-1. Two-ray model over flat earth.

The total E-field at the receiver is the sum of E-fields for the direct ray and for the reflected ray [11]. Assuming the distance the wave travels in the direct path and the reflected path are d_1 and d_2 respectively, and ground reflection is perfect so that the reflection coefficient is -1 and the reflected wave is 180^0 out of phase with the incident wave, the total E-field at distance d from the transmitter and time it is given by:

$$E(d,t)=\frac{E_0 d_0}{d_1}\cos\left[\omega_c\left(t-\frac{d_1}{c}\right)\right]-\frac{E_0 d_0}{d_2}\cos\left[\omega_c\left(t-\frac{d_2}{c}\right)\right] \qquad \text{Eqn. 2-2}$$

where E_0 is the free space E-field at distance d_0 from the transmitter, ω_c is the radian frequency of the wave, and c is the speed of light. The total E-field is approximate as

$$E(d,t) \approx \frac{2E_0 d_0}{d} \cdot \frac{2\pi h_t h_r}{\lambda d}$$

and the received power is

$$P_{r(d)} = \frac{P_t G_t G_r h_t^2 h_r^2}{d^4} \qquad \text{Eqn. 2-3}$$

where,

$P_r(d)$ is the receive power given a transmitter-receiver separation of d,

P_r is the transmit power,

G_t is the gain of the transmitting antenna,

G_r is the gain of receiving antenna,

h_t is the height of the transmitting antenna above ground,

h_r is the height of the receiving antenna above ground, and

d is the distance between the transmitter and the receiver.

As seen from equation 2-3 at large distance $\left(d >> \sqrt{h_t h_r}\right)$, the received power falls off with distance raised to the fourth power, or at a rate of 40 dB/decade. Path loss is much more rapid than that experienced in free space. Note also that at large values of d, the received power and path loss become independent of frequency. The path loss for the two-ray model can be expressed in dB as

$$PL(dB) = 40\log(d) - (10\log(G_t)) + (10\log(G_r)) + 20\log(h_t) + 20\log(h_r). \qquad \text{Eqn. 2-4}$$

The above equation shows a faster power loss (d^{-4}) than the free space model (d^{-2}) as distance increases. However, the two-ray model does not give a good result for a short distance due to the oscillation caused by the constructive and destructive combination of the two rays. Instead, the free space model is still used when d is small.

3. The shadowing model

The free space model and the two-ray model predict the received power as a deterministic function of distance. They both represent the communication range as an ideal circle. In reality, the received power at certain distance is a random variable due to multipath propagation effects, which is also known as fading effects. In fact, the above two models predict the mean received power at distance d. A more general and widely-used model is called the shadowing model [12].

The shadowing model consists of two parts. The first one is known as path loss model, which also predicts the mean received power at distance d, denoted by $\overline{P_r(d)}$. It uses a close-in distance d_0 as a reference. $\overline{P_r(d)}$ is computed relative to $P_r(d_0)$ as follows,

$$\frac{P_r(d_0)}{\overline{P_r(d)}} = \left(\frac{d}{d_0}\right)^{\beta}. \qquad \text{Eqn. 2-5}$$

β is called the path loss exponent, and is usually empirically determined by field measurement. From the free space model, we know that $\beta=2$ for free space propagation. Table 2-1 gives some typical values of β. Larger values correspond to more obstructions and hence faster decrease in average received power as distance increase. $P_r(d_0)$ can be computed from the free space model.

The path loss is usually measured in dB. So from Eqn. 2-5 we have

$$\left[\frac{\overline{P_r(d)}}{P_r(d_0)}\right]_{dB} = -10\beta \cdot \log\left(\frac{d}{d_0}\right) . \qquad \text{Eqn. 2-6}$$

The second part of the shadowing model reflects the variation of the received power at a certain distance. It is a log-normal random variable, that is, it is of Gaussian distribution if measured in dB. The overall shadowing model is represented by

$$\left[\frac{\overline{P_r(d)}}{P_r(d_0)}\right]_{dB} = -10\beta \cdot \log\left(\frac{d}{d_0}\right) + X_{dB} \qquad \text{Eqn. 2-7}$$

where X_{dB} is a Gaussian random variable with zero mean and standard deviation σ_{dB}. σ_{dB} is called the shadowing deviation, and is also obtained by measurement. Table 2-2 shows some typical values of σ_{dB}. Eqn. 2-7 is also known as a log-normal shadowing model. The shadowing model extends the ideal circle model to a richer statistic model: nodes can only probabilistically communicate if they are near the edge of the communication range.

Environment		
Outdoor	Free space	2
	Shadowed urban area	2.7 to 5
In building	Line-of-sight	1.6 to 1.8
	Obstructed	4 to 6

Table 2-1. Some typical values of path loss exponent, σ_{dB}.

Environment	(dB)
Outdoor	4 to 12
Office, hard partition	7
Office, soft partition	9.6
Factory, line-of-sight	3 to 6
Factory, obstructed	6.8

Table 2-2. Some typical values of shadowing deviation.

B. Link-layer protocols [13]

The transmission of information over sensor nodes link always results in some degradation in the quality of the information, because of limited bandwidth of the wireless channel combined with radio propagation loss. This means communication is inherently unreliable. Link-layer protocols are used to add information bits to the data bits to protect them against transmission channel errors. Coding of data provides a means of detecting errors at the receiving nodes. Error-detecting codes allow the presence of one or more errors in a block of data bits to be detected. Error-correcting codes allow the receiving terminal equipment to locate and correct a limited number of errors in a block of data. When error detection is employed, some form of retransmission scheme is needed so that the data block can be sent again when it is found to be in error. Forward error correction (FEC) provides a means of both detecting and correcting errors at the receiving node without retransmission of data. FEC codes add redundant parity check bits to the data bits in a way that allows errors to be located within a codeword. The two basic types of channel coders for FEC are block coders and convolutional coders.

1. Block codes

Block codes are codes in which there are k bits of message and (n-k) redundant check bits generated from the k message bits by a predetermined rule. In a systematic linear block code the first k bits of the codeword are the message and the remaining (n-k) bits are the check bits. A codeword with n bits of which k bits are data is written as (n, k). The code rate is $R = k/n$. The block codes have the capability of locating errors in a received codeword. Generally, a larger number of errors can be detected than can be located, but when an error can be located within a codeword, it can be corrected. When multiple errors have to be corrected, the procedure becomes extremely cumbersome. Binary cyclic codes (e.g. BCH) are one of block codes which implements very simple error-correction logic. BCH are popular linear block codes for long code words or data blocks. They can be generated and detected using shift registers and logic gates.

2. Convolutional codes

Convolutional encoding of data is performed by convolving k bits of the input with n generator polynomials to produce a rate k/n code, where $k \leq n$. Convolutional codes are generated by a tapped shift register and two or more modulo-2 adders wired in a feedback network. As each incoming information bit propagates through the shift register, it influences several outgoing bits, spreading the information content of each data bit among several adjacent bits. An error in any one output bit can be overcome at the receiver without any information being lost. If $k = 1$, the input stream can be continuously fed into the shift registers and the output can be continuously read at n/k times the rate of the input. Convolutional codes have the nice property that several different rate codes can be achieved using the same mother code, so a single hardware implementation can produce varying amounts of protection.

C. Media Access Control (MAC) protocols

Multiple access is the ability of a large number of users to simultaneously interconnect through a shared channel. There are 2 fundamentally different ways to share the wireless channel bandwidth among different nodes, fixed-assignment channel-access methods and random access methods. Fixed-assignment MAC protocols allocate each user a given amount of bandwidth, either slicing the spectrum in frequency (FDMA), time (TDMA), code (CDMA), or space (SDMA). Random-access are contention-based schemes, where users that have information to transmit must try to obtain bandwidth while minimizing collisions with other user's transmissions, such as IEEE 802.11, carrier sense multiple access (CSMA), multiple access collision avoidance (MACA) and MACA for wireless (MACAW). When all users do not have much data to send, random-access methods are more efficient than fixed-assignment methods, but they may suffer from collisions of the data, as all users are contending for the resources. MAC protocols can be evaluated in terms of energy dissipation, fairness and throughput.

1. FDMA

In frequency division multiple access, a given frequency band is divided and allotted to different channels. Each transmitting user is allocated a fixed carrier frequency. In receiving, since each user is operating in a fixed, carrier known frequency, different users can be separated, although they may work simultaneously. The receiving user can be tuned to the desired transmitting user, which will be selected and at the same time, other transmitting users will be rejected. Thus the FMDA technique is based on the principle of frequency translation; by the process of modulation, it is possible to shift or translate the frequency spectra of signals to various frequency ranges so that when they are transmitted over a single channel, they do not interfere with each other, and can be easily separated at the receiving end. A general case of FDMA is shown in Figure 2-2.

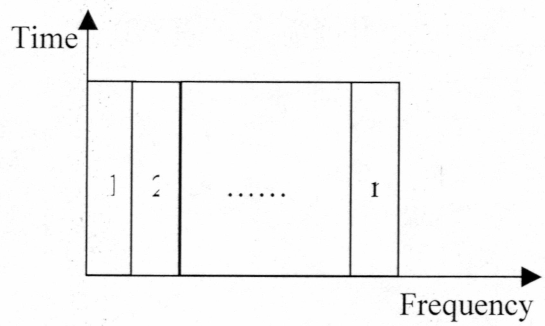

Figure 2-2. FDMA scheme.

FDMA offers a number of other advantages. FDMA's technology, which separates users in frequency, ensures that there is not interference from other simultaneous transmissions. FDMA is simple algorithmically from a hardware standpoint. FDMA technology does not need time synchronization. FDMA is fairly efficient when the number of users is small and the traffic is uniformly constant.

One of the disadvantages of FDMA is that it is not conducive to varying users; another problem with FDMA is if traffic is bursty, bandwidth is wasted, and at the same time, interfrequency protection bands waste bandwidth.

An analogy will be used to illustarate FDMA. Think of a noisy cocktail party where it gets harder to hear a conversation. The FDMA host solves divide the room into many compartments and allow only two guests to use each compartment at a time. The two guests have continuous access to their compartment, but not very many people can come to the party.

2. TDMA

Time division multiple access (TDMA) is a transmission technology that allows a number of users to access a single radio-frequency (RF) channel without interference by allocating unique time slots to each user within each channel.

TDMA relies upon the fact that the information has been digitized; that is, divided into a number of packets, typically milliseconds long. It allocates a single frequency channel for a short time and then moves to another channel. The digital samples from a single transmitter occupy different time slots as shown in figure 2-3.

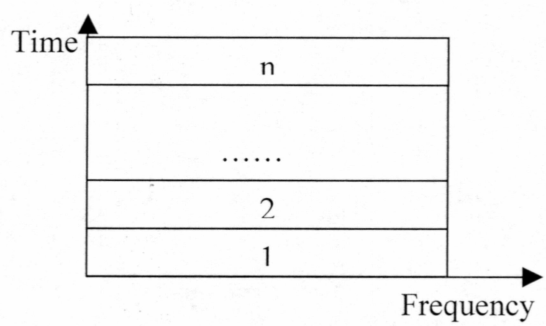

Figure 2-3. TDMA scheme.

In addition to increasing the efficiency of transmission, TDMA offers a number of other advantages. TDMA's technology, which separates users in time, ensures that there is not interference from other simultaneous transmissions. TDMA also consumes lower energy since the user is only transmitting a portion of the time.

One of the disadvantages of TDMA is that each user has a predefined time slot. When users do not have data to send, the time slot remains unused and this scheme is inefficient. Another problem with TDMA is that it is subject to multipath distortion. A signal coming from a tower to a handset might come from any one of several directions. It might have bounced off several different buildings before arriving which can cause multipath interference, which is more problematic for higher data rates, and short transmission time systems.

TDMA is like a cocktail party where multiple pairs of guests have access to the room, but only one pair can use the room at a time. They must then leave and allow another couple to enter. Throughout the evening all the guests rotate using the room.

Protocols often use a hybrid approach, for example, using FDMA and TDMA by allocating certain time and frequency slot for each user.

3. CDMA

CDMA is a spread-spectrum technology that allows multiple frequencies to be used simultaneously, as shown in figure 2-4. CDMA codes every digital packet it sends with a unique key. A CDMA receiver responds only to the desired key and can pick out and demodulate the associated signal.

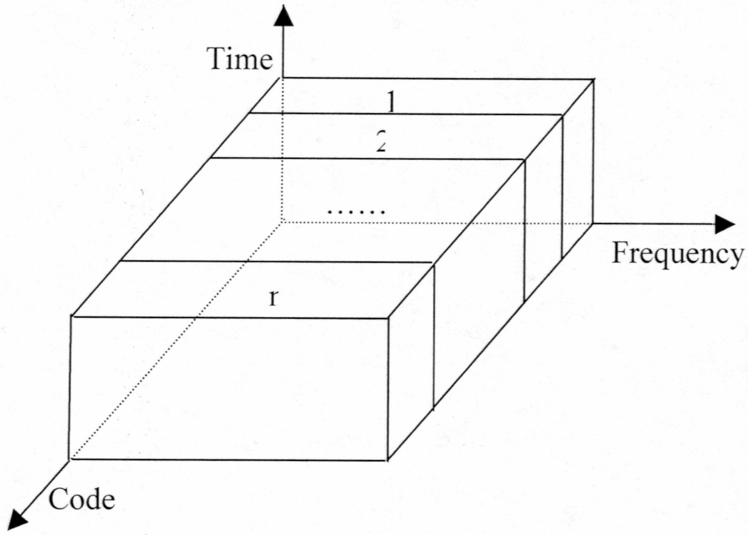

Figure 2-4. CDMA scheme.

CDMA uses unique codes to spread the baseband data in frequency before transmission. The signal is transmitted in a channel. The receiver then uses a correlator to de-spread the wanted signal, which is passed through a narrow bandpass filter. Unwanted signals will not be dispread and will not pass through the filter. Codes take the form of a carefully designed one/zero sequence produced at a much higher rate than that of the baseband data. The rate of a spreading code is referred to as chip rate rather than bit rate. Power control is very import in CDMA systems.

Of the different spread spectrum techniques, direct sequence is the most popular. A large problem with multi-access direct sequence spreading is the so-called near-far effect which is illustrated in figure 2-5. This effect is present when a CDMA interfering transmitter ("B" in figure 2-4) is much closer to the receiver ("C") than the intended transmitter ("A"). Although the cross-correlation of ``code A" and ``code B" is low, the correlation of the received signal from the interfering transmitter with ``code A" in the receiver can exceed the correlation of the received signal from the intended transmitter and the correct code. This signals from "node B" have the effect of raising the noise level. As a result proper data detection is hardly possible.

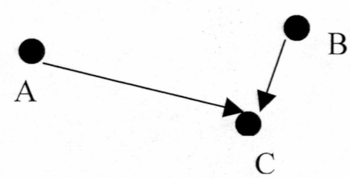

Figure 2-5. Near-far effect illustrated.

One of the main advantages of CDMA systems is the capability of using signals that arrive in the receivers with different time delays. This phenomenon is called multipath. Spread-spectrum techniques may suffer from interference among the users all of whom are on the same frequency band and transmitting at the same time. The unwanted signals of other users raise the noise floor in the frequency band. This limits the amount of users in the same band.

CDMA is like a cocktail party where everyone is in the same room talking at the same time. However, each pair of guests uses a different language to communicate. Using different languages makes it very easy for the guests to tune in to what the person speaking their language is saying and tune out everyone else.

4. **Pure Aloha [11]**

With Pure Aloha, users are allowed access to the channel whenever they have data to transmit. Because the threat of data collision exists, each user must either monitor its transmission on the rebroadcast or await an acknowledgment from the destination station, as shown in figure 2-6. By comparing the transmitted packet with the received packet or by the lack of an acknowledgement, the transmitting user can determine the success of the transmitted packet. If the transmission was unsuccessful it is resent after a random amount of time to reduce the probability of re-collision.

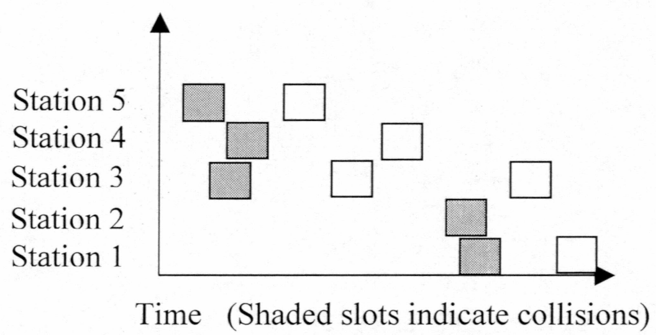

Figure 2-6. Pure Aloha Protocol.

Pure Aloha is superior to fixed assignment when there is a large number of bursty users. At the same time, Pure Aloha can adapt to a varying number of users.

One of the disadvantages is that its throughput is limited--theoretically proven throughput maximum of 18.4%. Another disadvantage is that it requires queueing buffers for retransmission of packets.

5. **Slotted Aloha [11]**

By making a small restriction in the transmission freedom of the individual stations, the throughput of the Aloha protocol can be doubled. Assuming constant length packets, transmission time is broken into slots equivalent to the transmission time of a single packet, as shown in figure 2-7. Stations are only allowed to transmit at slot boundaries. When packets collide they will overlap completely instead of partially. This has the effect of doubling the efficiency of the Aloha protocol and has come to be known as Slotted Aloha.

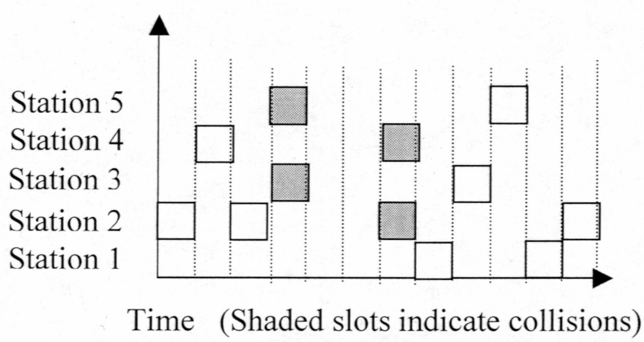

Figure 2-7. Slotted Aloha Protocol.

Slotted Aloha doubles the efficiency of Aloha while it still adapts to a varying number of users. Different from Pure Aloha, Slotted Aloha requires synchronization.

6. CSMA [11]

CSMA listens to the channel before engaging in transmission, thus greater efficiencies may be achieved. If the channel is idle, the data is transmitted. If the channel is busy, transmission is deferred. Persistent CSMA retries transmission immediately with probability p when the channel becomes idle. This may cause instability. Non-persistent CSMA retries transmission after a random interval.

7. MACA [4]

MACA (multiple access collision avoidance) was the first modern protocol which used RTS/CTS (request to send / clear to send) exchange and underscored the benefit of it over the then existing protocols (which were largely CSMA/CD based). The motivation was the hidden terminal problem. In MACA before a user sends data it sends an RTS message to the receiver. On success the receiver responds with a CTS. The nearby stations are also listening to this exchange. If a user hears RTS it waits for the corresponding CTS. If it does not hear CTS, it means any transmission it has will not interfere with the receiver. The assumption here is if you cannot hear the receiver, the receiver cannot hear you too. This helps alleviate the exposed terminal problem.

Any user, other than the original RTS sender, on hearing CTS will defer its transmission. The time for which to defer transmission depends on the packet length to be transmitted which is contained in the CTS packet. This takes care of the hidden terminal problem. Binary exponential backoff is used in case of collisions of RTS packets. MACA requires much simpler hardware because of absence of carrier sense.

8. MACAW [4]

Various practical problems with MACA were identified by MACAW (MACA for Wireless) and proposed changes solved some of them. This was one of the first wireless MAC protocols that were designed with fairness in mind. MACAW gets rid of Ethernet like unfairness associated with binary exponential backoff algorithms by proposing a copying form of backoff counter in which nodes use the backoff counter of a successful transmission to contend fairly in the next cycle. Also separate backoff parameters were introduced (corresponding to different streams) to avoid having this copied parameter to spread widely, even to areas with no congestion. It also proposed a multiple stream model for fairness among streams emerging from the same station. MACAW acknowledged the importance of link layer acknowledgements and made the protocol from RTS-CTS-Data to RTS-CTS-Data-ACK. The introduction of this ACK (acknowledgement) packet means that exposed terminals should not transmit now, or else they will trash the incoming ack. There are two ways of dealing with this, carrier sense or an explicit packet specifying the length of the transmission at the start of it. MACAW takes the latter approach to keep the hardware simple and calls this packet DS (data sending). Another control packet RRTS (Request for RTS) was added to let the receiver contend with the sender to improve fairness in cases when there are two receivers in the vicinity of each other (thus only one can receive). By making the protocol significantly more complex MACAW lost performance when the channel was lightly loaded but led to much better throughput and fairer allocation in presence of high loads.

D. Routing Protocols[4], [13]

1. Introduction

Protocols can be classified in four categories:

- Centralized or distributed
- Adaptive or static
- Demand-based or Table-driven or hybrid
- Multi-Hop or Clustering

When a routing protocol is centralized, all decisions are made at a center node, where as in a distributed routing protocol, all nodes share the routing decision. An adaptive protocol may change behavior according to the network status, which can be congestion on a link or many other possible factors. Associatively Based Routing (ABR) is an adaptive protocol, where associativity is related to spatial, temporal and connection stability of a mobile host. In table-driven approaches, every user continuously maintains the complete routing information of the network. When a user needs to forward a packet, the route is readily available; thus there is no delay in searching for a route. However, for a highly dynamic topology, the proactive schemes spend a significant amount of scarce wireless resource in keeping the complete routing information correct. Distance Vector Routing and Link State Routing belong to this kind. Demand-based nodes only maintain routes to active destinations. A route search is needed for every new destination. Therefore, the communication overhead is reduced at the expense of delay due to route search. Dynamic Source Routing (DSR) is this kind scheme. Hybrid methods use elements of both to come up with a more efficient one. Zone routing protocol is an example of a hybrid method. In multi-hop routing, a user delivers a packet to a destination through other peer users. On the other hand, in clustering approaches, the network is partitioned into clusters and a cluster head is elected in cluster based algorithms.

2. Destination Sequenced Distance Vector (DSDV)

DSDV is based on the idea of the classic Bellman-Ford routing algorithm. In DSDV, each node maintains a routing table containing the next-hop information for each reachable destination. Each entry has a sequence number, and if a new entry is given, it prefers the one with the greatest sequence number, or if their sequence is the same, it chooses the metric with the lowest value. Each node advertises an increasing even sequence number for itself. When Node A determines that destination Node D is unreachable, it advertises the next odd sequence for the route that has failed with an infinite metric count number. Any node receiving this infinite metric count updates its table for the matching route and waits until a greater sequence number with non-infinite metric count is received.

Since periodic updates are needed, it takes some time for the routing protocol to converge for building up a usable route. Those updates take quite a lot bandwidth. To reduce this amount, two types of information exchange are used, time-driven and event-driven. A full dump results in whole information broadcast, an incremental dump causes only differences of the last update to be broadcast. In a fast changing network, incremental packets can grow big, so full dumps will be more frequent. Triggered updates are also added to this protocol for adapting to ad-hoc networks.

3. Global State Routing (GSR)

In GSR, Each Mobile Host has knowledge of full network topology and it maintains a Neighbor list, a Topology table, a Next Hop table and a Distance table. The neighbor list of a node contains the list of its neighbors. For each destination node, the Topology table contains the link state information as reported by the destination and the timestamp indicating the time the destination has generated this information. For each destination, the Next Hop table contains the next hop to forward packets to for this destination. The Distance table contains the distances of shortest path to each destination node. Additionally, a weight function is used to compute the distance of a link. If min-hop shortest path is the objective, this weight function returns 1 if two nodes have direct connection. This weight function may also be replaced with other functions for routing with different metrics.

4. Dynamic Source Routing (DSR)

DSR is a source-routed on-demand routing protocol. Every node maintains a route cache containing the source routes that it is aware of. The node updates the entries in the route cache if there is a better route, when it learns about new routes.

DSR requires that each packet keep its route information, thus eliminating the need for every node in the network to do periodic route discovery advertisements. DSR performs a route discovery and takes required actions for maintaining that route.

The disadvantage of this approach is that the protocol has a major scalability problems due to the nature of source routing. The packets may be forwarded along stale cached routes.

5. The Ad-Hoc On Demand Distance Vector (AODV)

AODV is a combination of both DSR and DSDV protocols. It has the basic route-discovery and route-maintenance of DSR and uses the hop-by-hop routing, sequence numbers and beacons of DSDV. The node that wants to know a route to a given destination generates a ROUTE REQUEST. The route request is forwarded by intermediate nodes that also create a reverse route for itself from the destination. When the request reaches a node with a route to the destination it generates a ROUTE REPLY containing the number of hops requiring to reach destination. All nodes that participate in forwarding this reply to the source node create a forward route to the destination. This state created from each node from source to destination is a hop-by-hop state and not the entire route as is done in source routing.

III. The Protocol Stack

A. Assumptions

The following assumptions are made in the sensor network examined in this thesis.

1. Each node has a global ID and starts with an equal amount of energy.
2. Each node is stationary (does not move), but networks are dynamic: some nodes may die or be moved, and some new nodes may be added into the net.
3. Each node can change its transmission power. There are two transmission power levels. The low transmission power level is used for contact within a cluster. The high transmission level is used to communicate with neighbor clusters. The range of the high level is at least 3 times that of the low level.
4. A message sent by a node is received correctly within a finite time by all of its relative neighbors, such as 1-hop neighbors for low level, 3-hop neighbors for high level.
5. Network topology does not change during the algorithm execution.
6. Data rate is not high.

B. Overview

This is a cluster-based multi-hop network protocol stack. As show in figure 3-1, sensor nodes are grouped into clusters. In a cluster, every node can communicate with each other directly in low transmission power, so every node can act as cluster head by turn. Members in a cluster contact only the cluster head. They send data to the cluster head under TDMA MAC method, so member may go to sleep state in many slots to save energy cost. The cluster head aggregates data before transmitting it out. The cluster head transmits its data to a neighbor cluster with higher transmission power, which is "nearer" (has less hops) to the base station. To reduce inter-cluster interference, each cluster communicates using direct-sequence spread spectrum (DS-SS). Each cluster uses a spreading code. Because inter-cluster communication uses higher transmission power than intra-cluster communication, the inter-cluster communication may interfere with intra-cluster communication badly if those two kinds of communication happen at the same time. It is necessary to separate the two communication periods. Both intra-cluster and inter-cluster communication periods are divided into slots, as shown in figure 3-2. The intra-cluster slots are used to exchange data between cluster members. The inter-cluster slots are used for inter-cluster communication. If a cluster wants to send data through its neighbor cluster, it should reserve a slot from that neighbor cluster. For example, in figure 3-1, if cluster C wants to forward data through cluster B, it should reserve a slot from cluster B while cluster B has to reserve slots from cluster A in order to send data to the base station. All cluster heads use the same slot dividing method. Therefore, synchronization is needed.

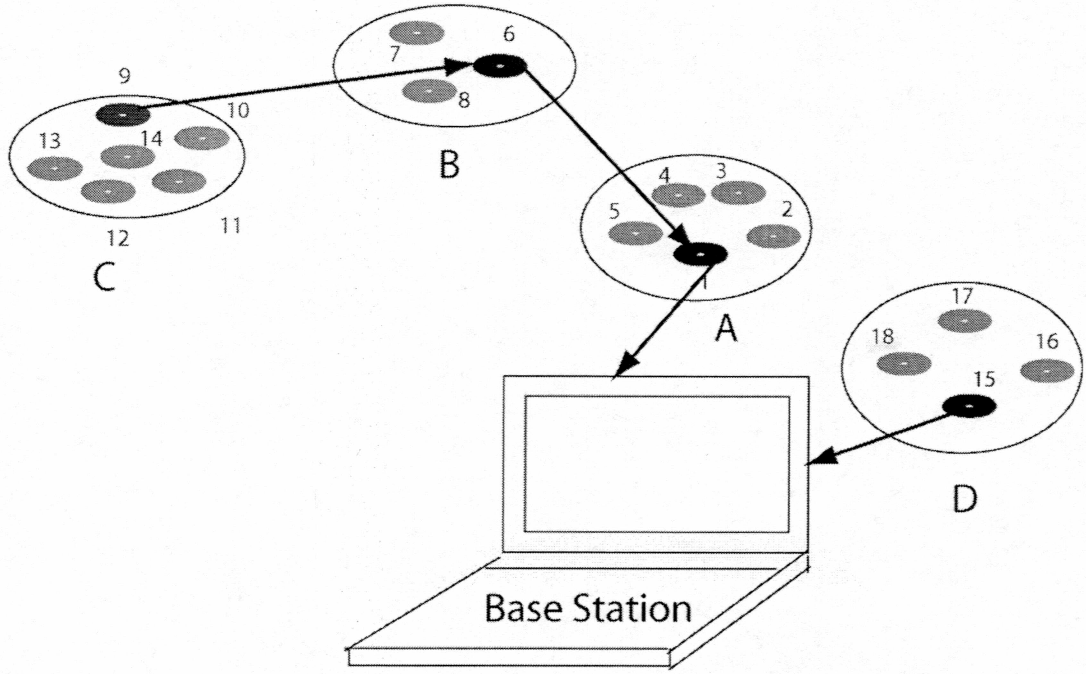

Figure 3-1. Cluster based multihop scene.

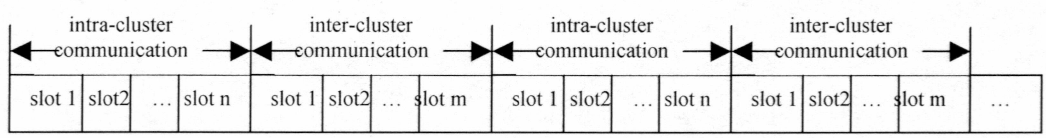

Figure 3-2. Two kind of communication slots.

C. Clustering Protocols

The algorithm produces clusters of nodes where all members of the cluster can communicate with each other directly. In addition, each element is required to be a member of exactly one group. The tighter coupling between the cluster members eliminates the dependency and associated bottlenecks of having one group leader for intra-group communications. Any member can be the leader and the failure of individual members does not affect the connectivity within the cluster.

1. Phase I -- to know neighbors

In this stage, every sensor node will get the IDs of its 1-hop neighbors. Every sensor node broadcasts what it has heard only once by a non-persistent carrier-sense multiple access MAC method. After being deployed, each sensor node keeps listening. If a channel is available, it broadcasts what it has heard. After broadcasting, the sensor node keeps listening to its neighbors. The transmission power level used in this stage is low (intra-cluster), because only 1-hop neighbors are supposed to hear this kind of broadcasting. The state chart of a sensor node is shown in figure 3-3.

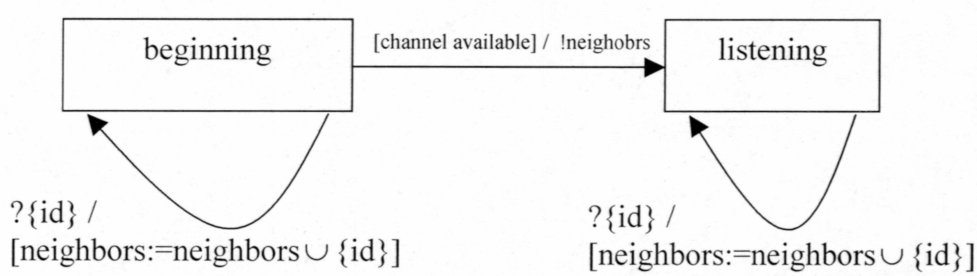

Figure 3-3. State chart of knowing neighbors.

Lemma 1: By the end of this stage, each node knows its 1-hop neighbors.

Proof: For any given sensor node, all of its neighbors will broadcast once, so it will hear all of its neighbors, no matter who broadcasts first.

Lemma 2: A sensor node's broadcast neighbors may be fewer than its real neighbors.

Proof: After hearing a broadcast, a sensor node will broadcast to its neighbors from which it has heard if a channel is available. A sensor node may broadcast before it hears from all of its neighbors.

Theorem 1: By the end of this stage, a sensor node may not get complete information about its neighbors' neighbors, but it can know if its neighbors can contact with each other.

Proof: From Lemma 2, a sensor node's broadcast neighbors may be fewer than its real neighbors, so a sensor node may not get complete information about its neighbors' neighbors. From Lemma 1, a sensor node can hear from all of its neighbors. If two neighbors can hear from each other, the first one broadcast does not know the second one when it broadcasts, but the second one has heard the first one before it broadcasts, so the sensor node knows this contact between its two neighbors.

Here is an example network.

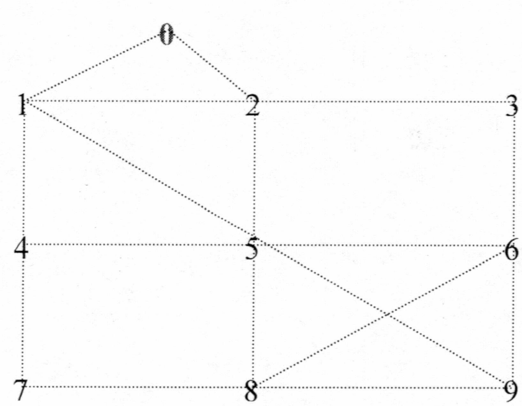

Figure 3-4. A 9-node simple sensor network.

In this example, there is one base station and 9 sensor nodes. The line between two nodes means a 1-hop contact (in low transmission power). For example, node 3 can contact with nodes 2 and 6 in low transmit level while node 5 can hear nodes 1, 2, 4, 6, 8 and 9.

node	Neighbors of broadcast	Content of hearing	
		Before broadcast	After broadcast
1	0	0	2: {1}; 4: {1}; 5: {1;2;4;6}
2	{1}	1: 0	3:{2}; 5:{1;2;4;6}
4	{1}	1: 0	5:{1;2;4;6}; 7:{4}
3	{2}	2:{1}	6:{3}
6	{3}	3:{2}	5:{ 1;2;4;6}; 8:{ 5;6;7}; 9:{ 5;6;8};
5	{1;2;4;6}	1: 0; 2: {1}; 4: {1}; 6: {3}	8: {5;6;7}; 9: {5;6;8};
7	{4}	4: {1}	8: {5;6;7}
8	{5;6;7}	5: {1;2;4;6}; 6:{3}; 7:{4}	9: {5;6;8}
9	{5;6;8}	5: {1;2;4;6}; 6: {3}; 8: {5;6;7}	

Table 3-1. The first phase communication of the 9-node simple sensor network.

We assume node 1 broadcasts first. Node 1 broadcasts that it can hear nothing (Lemma 2). Then node 2 broadcasts that it can hear only node 1. After broadcasting, they all keep listening. So after node 2 broadcasts, node 1 knows it can hear node 2. It will hear node 4 and node 5 finally. Thus far, it knows all of its neighbors (Lemma 1). When node 1 hears from node 2 and node 4, which tell it that they can only hear node 1, node 1 can not get complete information about node 2 and node 4's neighbors (Theorem 1, part 1). When node 1 hears node 5, which can hear nodes 1, 2, 4 and 6, node 1 knows that among its neighbors, node 2 and node 5, node 4 and node 5 can hear each other (Theorem 1, part 2). The sample broadcast sequence is listed in table 3-1, where " 2:{1}" is the content of node 2's broadcast, " I am node 2, I can hear only node 1 so far".

2. **Phase II -- set up cluster and routing [14], [15]**

In this stage, sensor nodes are grouped into clusters. Routing will set up while clusters are set up.

(a) About a cluster

Definition: a cluster is a group of sensor nodes in which all members can communicate with each other directly in low transmission power.

Theorem 2: By the end of phase I, every node knows with whom it can comprise a cluster.

Proof: From theorem 1, each node knows if its neighbors can contact each other, so it can pick up a sub set where all members of the set can communicate with each other directly.

For example, node 1 knows that its neighbors include nodes 2, 4 and 5, so it can comprise a group with node 2, 4 or 5 respectively. At the same time, it hears from node 5 that node 5 can communicate with node 2 and node 4 respectively. Node 2 and node 4 can not communicate with each other directly, so possible clusters could be {1, 2, 5}, {1, 4, 5}, {1, 2}, {1, 4}, {1, 5}. But, which clusters will be chosen depends on the clustering algorithm.

(b) The clustering algorithm

The objective of the clustering algorithm is to find interconnected set of clusters covering the entire node population. Namely, the system topology is divided into small partitions (clusters) with independent control. A good clustering scheme will tend to preserve its structure when a few nodes are moving and the topology is slowly changing. Otherwise, high processing and communications overhead will be paid to reconstruct clusters. Within a cluster, it should be easy to schedule packet transmissions. For the sake of efficiency, clusters should be as big as possible. Finding the maximum cluster size can be formulated as a maximum clique problem [16].

Let G(V, E) be a simple undirected graph where V is set of vertices, and E is set of edges. The adjacency matrix of G is a matrix $A_G=(a_{ij})_{n\times n}$, where $a_{ij}=1$ if $(i, j)\in E$, and $a_{i,j}=0$ if $(i, j) \notin E$. The neighborhood of the vertex i is denoted by $N(i) = \{j\in V: (i, j) \in E\}$. According to the definition of a cluster, any member in a cluster is a neighbor of others. This kind of cluster is called clique in graph theory. A clique Q is a subset of V such that any two vertices of Q are adjacent. The maximum clique problem asks for a clique of the maximum cardinality. This cardinality is called the clique number of the graph. The maximum clique problem is NP-hard, that means that it is considered unlikely that an exact polynomial time algorithm for its solution exists. Approximation of large cliques is also hard. To practically find a possible good substitution for the maximum clique, many heuristic methods were developed. We use a simple algorithm. First we choose a node that has maximum vertex degrees as a member of maximum clique. Then for its neighbor, we choose another node that has maximum vertex degrees among the first clique member's neighbors. We get the second clique member's neighbor set from the set of the first clique member. From the second clique member's neighbor, we choose the node with maximum vertex degrees as third clique member, and so on. The algorithm is shown in figure 3-5.

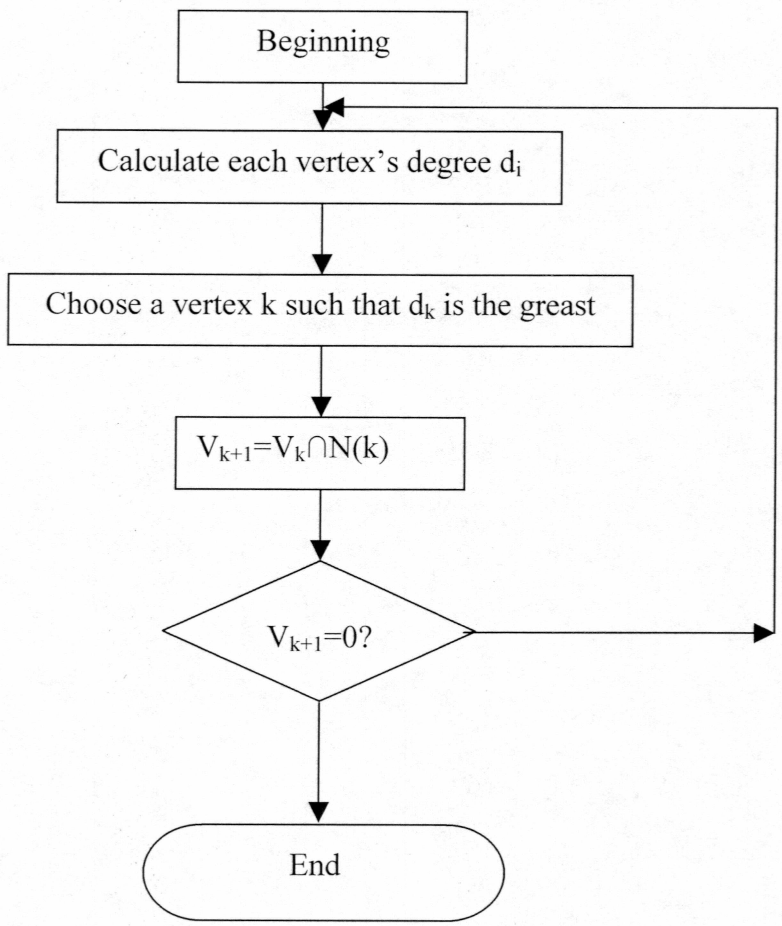

Figure 3-5. A simple maximum clique algorithm

There is an uncertainty if more than one vertex of the maximum degree exists. We suppose that v_k is chosen among them randomly in this case.

For example, in the previous example, node 6 is a neighbor of node 3, 5, 8, 9, as shown in figure 3-5. For node 6, N(6)={3, 5, 8, 9}

$$A_G = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix}$$, $d_3=0$, $d_5=2$, $d_8=2$, $d_9=2$. Nodes 5, 8, 9 have the same degree. We assume node 5 is chosen into the maximum clique first. Then Q={5}, node 5's neighbor set $V_2=\{8, 9\}$ $d_8=d_9=1$, we assume node 8 is chosen into maximum clique in the second turn, then Q={5, 8}, node 8's neighbor subset $V_3=\{9\}$. Finally, we get Q={5, 8, 9}, so the cluster comes out to be {6, 5, 8, 9}.

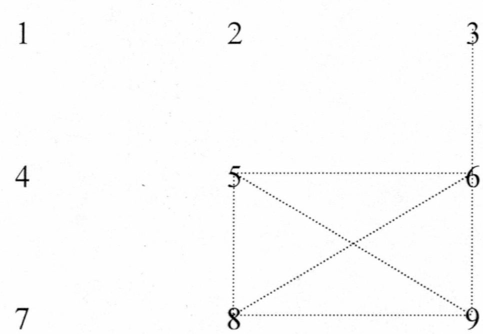

Figure 3-6. The neighbors of node 6.

Unfortunately, this algorithm may not always find out the maximum cliques. For example, for node 5, N(5)={1, 2, 4, 6, 8, 9}, $A_G = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 \end{pmatrix}$. $d_1=d_6=d_8=d_9=2$. If node 1 is chosen first, the resulting cluster will be {5, 1, 2}, where the maximum clique should be {5, 6, 8, 9}. To compensate this limit, we let the prospective cluster head wait for 10/|Q| milliseconds before broadcasting a recruit message. In this case, node 6 gets its prospective cluster of {5, 6, 8, 9}, so node 6 broadcasts a recruit message after 10/4=2.5 milliseconds. Node 5 get its prospective cluster of {5, 1, 2}; node 5 should wait for 10/3=3.3 milliseconds, so node 5 will be recruited before it begins to broadcast its recruit message. Node 5 need not recruit at all! This will be discussed further in the following section.

(c) Hop field based forwarding

In this protocol stack, no explicit forwarding path states are needed. Each cluster only needs to maintain the next hop from this cluster to the base station. Once the hop field is set up, any cluster can deliver the data to the base station. The backoff-based hop field setup algorithm establishes the optimal hop field in a single pass when there is no message loss or delay. At each cluster, the hop field is defined as the minimum hop from that cluster to the base station on the optimal path. The field has only one state at each cluster--the minimum hop count to base station. The hop field minimum forwarding is important if the user is interested in observation from multiple sensors. This way, we can effectively enable only the clusters along the minimum hop path to forward data without keeping any path states.

A straightforward solution to set up the hop field would be through flooding. Initially, after the base station broadcasts an ADV (advertisement) message containing its own hop(0), the message propagates throughout the network. Upon hearing an ADV message from cluster head M, cluster head N has a path with hop h_M+1, where h_M is cluster head M's hop. Cluster head N then compares its current hops to h_M+1. If the new hop is smaller, it sets its hops to h_M+1, and broadcasts an ADV message with its new cost. Whenever a cluster head receives an ADV message, leading to a smaller cost, it resets its cost and broadcasts a new ADV message. Eventually, every cluster head may calculate the optimal cost to the base station through flooding. The reason that a cluster head broadcasts more than once is that it broadcasts immediately after obtaining a lower cost, no matter whether the cost is optimal or not. If the broadcast is deferred at the cluster head to the time after this cluster head has heard the message leading to the minimum hop, the cluster head may broadcast only once, carrying its minimum hops. Thus how long the cluster head defers its broadcast become critical for reducing the tremendous message overhead. In a backoff-based approach, upon hearing a ADV message, a cluster head sets a backoff timer that expires after hop*x where x (x=10, for example) is a constant, and "Hop" is the hops contained in that ADV message. Using this method, each cluster head broadcasts once and only once with the optimal hop and suppresses other non-optimal advertisement messages. This method makes the total deferral time proportional to the optimal hop at a cluster head and establishes the minimum hop field with only one message broadcast at each cluster when there is no delay or message loss.

(d) To set up cluster and routing

At this stage, the non-persistent carrier-sense MAC method is used. The transmission power is at a high level. A cluster is identified by its neighbors through a spreading code. The base station broadcasts first. The nodes who hear the base station calculate their clusters using the "findmaximumclique" algorithm. The node who has the maximum cluster members will broadcast the recruiting message first, because every node will wait for 10/(number of prospective cluster members) millisecond, as discussed in the previous section. The recruit message includes the following information: cluster members, a spreading code, hops from base station (0 for the first cluster head), inter-cluster slot usage situation, next-hop cluster's unique spreading code (base station for the first cluster head), the number of slots the cluster want to reserve from the next-hop cluster (reserve a slot from base station for the first cluster head). Upon hearing the recruiting message, the nodes check recruited members. If it is recruited, it will stop the recruiting process and become a member of the cluster. If it is not recruited, but some of its neighbors are recruited, it will delete those recruited nodes from its neighbors and recalculate maximum cluster number, choose a spreading code and public slot different from its neighbor clusters (further discussion in the following section), reserve a slot from a nearer cluster, prepare to broadcast a recruiting message. There exists a problem. If two nodes get the same cluster number, but one can hear the base station (hop =1), another one can not hear base station, but can hear the first cluster head (hop =2), who should broadcast the recruiting message first? In order to set up hop field, the node with hop=1 should broadcast first. Thus far, a node must delay broadcasting for two reasons: delay 10*hops milliseconds to set up the hop field, and delay 10/(number of prospective cluster members) milliseconds to get maximum cluster. Combining those two backoffs gets hop*10+10/(number of cluster members) milliseconds total delay. This procedure is formulated as in the following.

Procedure cluster and routing setup
Data Structure
Neighbor(i) = = set of node i's neighbors
G(i) = = the graph consist of node i's neighbors and its edge.
Cluster(i) = = FindMaximumClique(G(i)).
Code(i) = = unique spreading code of cluster(i)
Hops(i) = = number of cluster (i) 's hops from base station
Slot-situation(i)(k) = = usage of k-th slot. 00: available, 01: public slot, 10: occupied, 11: occupied but not received correctly, the cluster that transmits in this slot should re-transmit the last data.
Slot(i)(k) = = (k-th slot, slot-situation(k)), it is descript of k-th slot of cluster (i))
Slots(i) = = {slot(i)(k)| k=1 to n}, it is descript of cluster(i)'s slot
Next-code(i) = = the spreading code of cluster (i)'s next hop cluster.
Next-slot(i) = = the no. of slot the cluster i want to reserve form next hop cluster
Hops(i) := 1
Next-slot(i) :=0
Recruit(i) = = (cluster(i), code(i), hops(i), slots(i), next-code(i), next-slot(i))
Begin
(1) receive recruit(j)
(2) if node i $\in$ cluster (j) go to end
(3) if cluster(i) $\cap$ cluster (j) $\neq$ 0, then Neighbor(i) := neighbor(i) \ (cluster(i) $\cap$ cluster(j)), cluster(i) := FindMaximumClique(G(i)).
(4) set time :=hops(j)*10 +10/|cluster(i)| millisecond.
(5) if hops(j)<hops(i)-1, then hops(i) :=hops(j)+1, next-code :=code (j), select next-slot(i) where slot-situation(j)(k) = = 00; select slot(i)(x) where slot-situation(j)(x) = = 00; slot-situation(i)(k) := 10, slot-situation(i)(x) := 10.
(6) if receive signal before time out, then go to (1)
(7) if channel available, then broadcast recruit(i)

(8) end
End

The following flow chart shows this process.

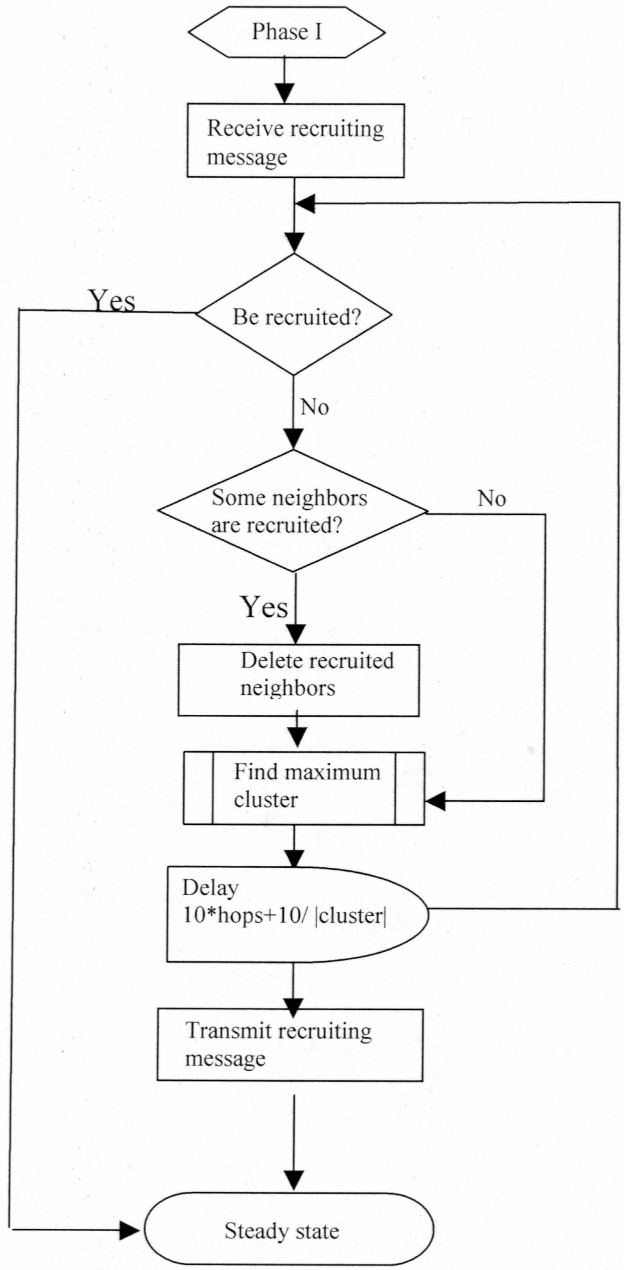

Figure 3-7. Set up cluster and routing.

In a sensor network scene such as figure 3-1, after the base station broadcasts, nodes 1-5 and nodes 15-17 prepare to broadcast the recruiting message, but node 1 gets the maximum prospective cluster, so it broadcasts first. Nodes 2-5 hear recruiting messages and find they are recruited, so they stop the recruiting process. Nodes 6-8 who have not heard the base station hear node 1's recruiting message too. They set their hop to 2. Node 6 gets 3 prospective cluster members, hence it delays 2*10+10/3 milliseconds, while node 15 with 3 prospective cluster member who can hear the base station (hop=1) will delay 1*10+10/3 milliseconds. So node 15 will broadcast first. This combining backoff method ensures setting up the hop field and getting maximum clusters. Node 15 chooses a spreading code and public slot different form node 1 and reserves a slot from the base station different from node 1, then broadcasts its recruiting message. Node 6 choose node 1's cluster as the next hop (upstream) cluster, and reserves a slot from node 1's cluster's slot, chooses a spreading code and public slot different form node 1, then broadcast its recruiting message. After that, node 12 will set up its own cluster in the same way.

D. Steady State Protocols [17]- [20]

1. Intra-Cluster communication

(a) Intra-cluster TDMA scheme

In a cluster, the cluster head node acts as a local control center to coordinate the data transmissions in its cluster. At end of phase II, the cluster head sets up a TDMA schedule and transmits this schedule to the nodes in the cluster. This ensures that there are no collisions among data messages and also allows the radio components of each non cluster head node to be turned off at all times except during their transmit time and cluster meeting time, thus minimizing the energy dissipated by the individual sensors. The cluster head keeps its receiver on to receive all the data from the members of the cluster. Once the cluster head receives all the data, it can perform data aggregation, and then send the resultant data to the next hop cluster at the reserved inter-cluster slot. After a slot for each node, there is an intra-cluster meeting slot and a free slot. In an intra-cluster meeting slot, the cluster head broadcasts the cluster's spreading code, cluster members, and nodes from which the head has not received data correctly. Intra-cluster synchronization will be implemented in an intra-cluster meeting slot too. In the free slot, the node whose data was not received by the cluster head correctly can retransmit its data. As shown in figure 3-8, the free slot can also be used to maintain the cluster. This will be discussed in the following section. The slotted Aloha MAC method is used in the free slot period.

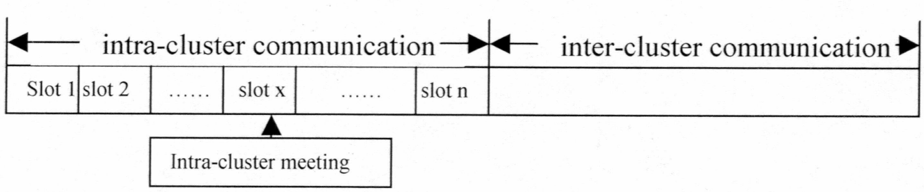

Figure 3-8. Intra-cluster slots.

(b) Spreading code [4]

Because radio is a broadcast medium, transmission in one cluster will affect communication in a nearby cluster. To reduce inter-cluster interference, each cluster communicates using direct-sequence spread spectrum (DS-SS). Each cluster uses a unique spreading codes, all the nodes in the cluster transmit their data to the cluster head using this spreading code and the cluster head filters all received energy using this spreading code. Since there is a small set of "good" spread spectrum codes which have low cross correlation, spatial reuse of codes will be important. Thus each cluster is assigned a single code, which is different from the codes used in the neighbor and the neighbor's neighbor clusters. For example, in figure 3-1, cluster A must use a code different from cluster B and cluster C. If cluster A and cluster C use the same code, cluster A may cause interference when cluster B sends data to cluster C. The problem of the code selection may be formulated as a graph coloring problem. For simplicity, we assume there is a pre-defined list. Clusters choose a spreading code from the list in turn. When a cluster head hears a neighbor's recruiting message that chooses the i-th spreading code in the list, it tries to use the (i+1)-th code in the list which is different from its neighbor and neighbor's next hop code.

(c) Head rotating

Having to receive data from all the nodes in the cluster, perform signal processing functions on the data, and transmit the data to the next hop cluster, a cluster head is much more energy intensive than a non cluster head node. To avoid overly-utilized nodes that will run out of energy before the others and evenly distribute the energy load among a cluster, each node should take its turn as cluster head. In this protocol stack, once the clusters are formed, there is no set-up overhead at the beginning of each round. In the cluster set up stage, a cluster head broadcasts its members. The first one is the head for the first round. The second node listed in the cluster becomes cluster head for the second round, and so forth.

To reduce energy dissipation, the cluster head uses power control to set the amount of transmission power based on the received strength of the next-hop cluster-head. For example, in figure 3-9, node a in cluster II can use less power when it communicates with node 4 in cluster I than communicating with node 2.

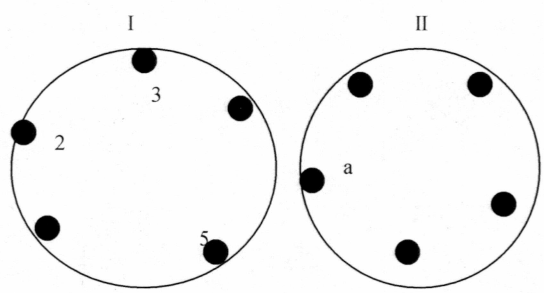

Figure 3-9. Transmission power sensitivity.

(d) Cluster maintenance

As these sensors are very tiny, they are vulnerable to failure or being accidentally moved. Hence, sensor networks should maintain network connectivity even if some of their sensors fail or are moved. For example, sensors located in a forest may be vulnerable to any kind of mobility (e.g., human, animal, insect, rain, or wind). Our solution is intra-cluster meeting broadcasting. In the intra-cluster meeting slot, the cluster head broadcasts the cluster's spreading code, cluster members, and nodes whose data has not been received by the head at all or incorrectly. If a node has not send data to the head for more than some times (5 for example), it will be removed from the cluster. Cluster meeting broadcasts are sent without a spreading code. The intra-cluster meeting broadcasting can also be used to recruit new members. The new node listens to the channel for a period of time. If it can hear all member nodes in the cluster, then it decides to join the cluster and uses free slots to transmit packets temporarily. The frame is readjusted after each member joins or leaves.

2. Inter-cluster communication

(a) Inter-cluster TDMA and slotted Aloha

As described in the previous section, inter-cluster communication is divided into slots also. There are three kinds of slots. The first kind of slot is reserved for fixed neighbors using TDMA MAC methods. The second kind of slot is used as the public slot. The third kind of slot is free slots using slotted Aloha MAC methods. At the end of the second phase, the up stream cluster head handles the down stream cluster's reservation, and makes a TDMA scheme for the down stream clusters.

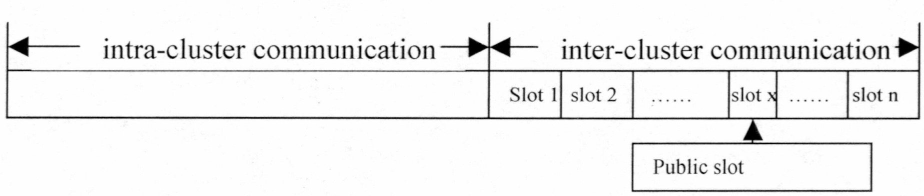

Figure 3-10. Inter-cluster slots.

In the public slot, the cluster head broadcasts the cluster members, the unique spreading code, and slot usage, which include every slot's situation, such as available (free slots), occupied and last data transmission correctly, occupied and last data transmission incorrectly, or used as the public slot. If a down stream cluster finds that its last data was not received correctly, it can re-transmit the last data in the up stream cluster's free slot using the slotted Aloha scheme. Public slot broadcast does not use a spreading code. So every node and cluster can hear. Neighbor clusters should choose different slots for the public slot. Synchronization is also made in the public slot. All the down stream clusters should be synchronized with the up stream clusters.

(b) Hop-by-hop error recovery [21]

According to this protocol stack, sensor networks rely on a multi-hop forwarding technique to exchange messages. As we know, sensor networks usually operate in harsh radio environments, so errors accumulate exponentially over multi-hops. Assume that the packet error rate of a wireless channel is p, then the chances of exchanging a message successfully across a single hop is (1-p). The probability that a message is successfully received across n hops decreases quickly to $(1-p)^n$. If the channel is highly reliable, for example, the error rate is less than 1%, the probability of a successful end-to-end delivery is well above 90%. However, sensor networks usually operate on low-power RF communication, which cannot rely on using higher power to boost the link reliability when operating under harsh radio conditions. Sometimes, the channel's error rate is in the range of 5%-10% or even higher. End-to-end error recovery is not a good candidate for reliable transport in wireless sensor networks. In hop-by-hop recovery, intermediate nodes also take responsibility for loss detection and recovery so reliable exchange is done on a hop-by-hop manner rather than an end-to-end one. This approach essentially segments multi-hop forwarding operations into a series of single hop transmission processes that eliminate error accumulation. The chance of exchanging a message successfully across a single hop is (1-p). Therefore, the probability of detecting loss in a negative acknowledgement system is proportional to (1-p), rather than decreasing exponentially with growing network size as in the case of end-to-end approaches.

(c) Routing maintenance

After setting up next hop routing, the cluster can use this up stream cluster to forward data to the base station. The change of the up stream cluster's head does not affect the data forwarding, so not much routing maintenance is needed.

If the whole up stream cluster dies, the cluster that uses this up stream cluster to forward data will lose connection temporarily. In this case, the cluster can use other neighbor clusters' free slot to forward data under the free slotted Aloha scheme temporarily, and eventually apply a fixed slot using a TDMA scheme.

E. Features and properties of protocol stack

The clustering and routing algorithm forms clusters and sets up routing by using a distributed algorithm, where nodes make autonomous decisions without any centralized controls. An advantages of this approach are that no long-distance communication with the base station is required. In addition, no global communication is needed to set up the clusters and routing. The clustering and routing scheme can preserve its structure when a few nodes are moving and the topology is slowly changing. The sensor network is scaleable under the protocol stack. The advantage of the protocol stack can be summarized as follows.

1. A TDMA scheme in a cluster allows non cluster head nodes to go to the sleep state, except in the time slot of translating data.

2. Nodes in a cluster take turns as head, so no single node will be drained of energy.

3. Because cluster members are tightly coupled, each sensor node can be cluster head in turn. There is no set-up overhead at the beginning of each round.

4. A cluster head need not remember the whole state of the sensor network. It only needs to remember its neighbor cluster.

5. Because the next hop is a whole cluster, it is more stable than a single node, so routing is stable. The cluster needs only to remember the spreading code of its next hop cluster, no matter who is the cluster head of the next hop cluster.

6. The sensor network is scalable. Error rate will not accumulate across the sensor network.

7. The intra-cluster communication (the low transmission power) and the inter-cluster communication (the high transmission power) are at different periods, so they will not interfere with each other though the latter one uses higher transmission power.

IV. Analysis

One of the operational challenges in nanosensor networks is its limited irreplaceable power resources. Energy costs for networks can be traced back to two main sources. One is the amount of energy required to set up routing and clusters. The second is the amount of energy required to detect phenomena and transport the data across the network. Estimating the first amount is very hard to formulate. In this section, we will discuss the energy consumed in nanosensor networks steady phase. By analyzing energy, we discuss sensor node deployment, optimization of cluster size, propagation condition effects and data correlation probability.

A. Sensor node deployment

Assuming we should monitor a circular area with the radius of R, where phenomena are uniformly distributed, how should sensor nodes be deployed? An intuitional answer is to locate the base station at the origin of the circle then deploy sensor nodes uniformly, as in figure 4-1. A matlab program is used to simulate a uniform distribution of sensor nodes in the field, given in appendix 1.

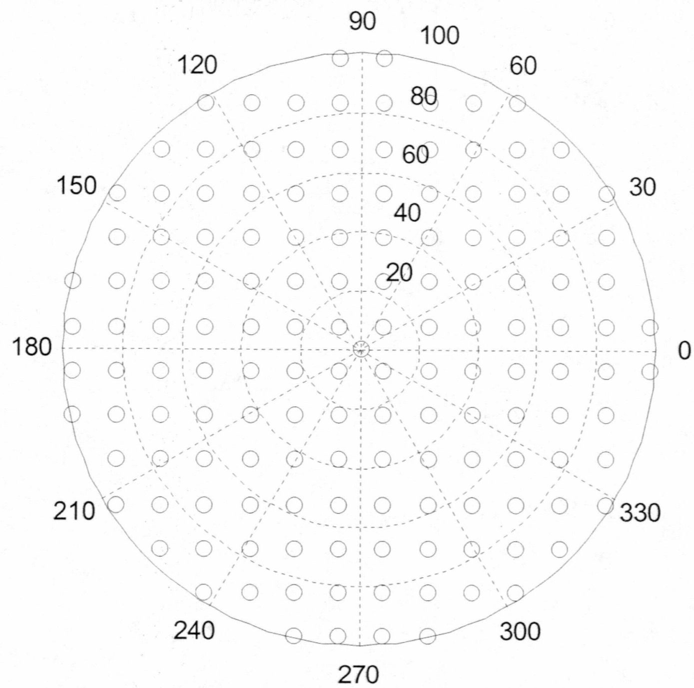

Figure 4-1. Simulation of uniform sensor nodes (150) deployment.

If sensor nodes are deployed in this way, sensor nodes that are near the origin should detect the phenomena, transmit their data and relay the data transmitted from farther sensor nodes towards the base station as well. The nearer sensor nodes are, the more energy they will consume, the earlier they will die. If near sensor nodes die, messages from farther nodes can not be relayed toward the base station. The whole sensor networks fail. This raise the question: Is there any optimization deployment way to let sensor nodes use up their energy at almost the same time?

In order to answer this, we need to analyze the energy cost density of the detection field. The energy cost density is the percentile of energy cost along the radius distance from the origin. As we discussed above, energy cost is not uniformly distributed. All energy comes from sensor nodes. In this abstract level, we regard sensor nodes as energy source particles. If it is reasonable to assume that each sensor node contains the same amount of energy, it is also reasonable to conclude that the energy cost density function f_e is equal to the sensor node density function f_s, $f_e = f_s$. Likewise are their distribution functions equal, $F_e = F_s$.

Assume sensor nodes are organized ideally as shown in figure 4-2. Each cluster has the same diameter (D, for example). The clusters whose distances from the origin are the same constitute an annulus. It is obvious that all sensor nodes in the same annulus consume the same amount of energy. Instead of studying single sensor nodes, we study sensor nodes in the same annulus together.

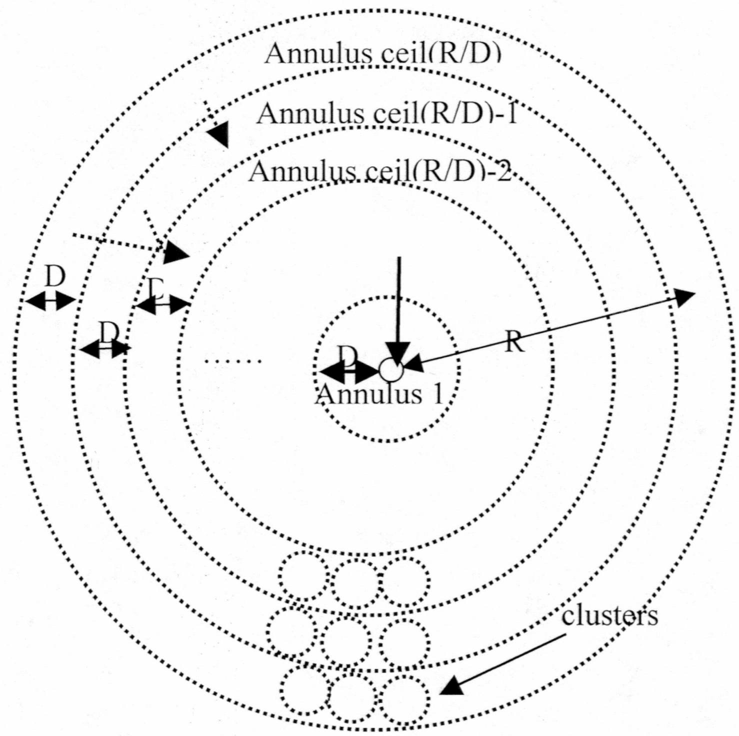

Figure 4-2. Calculation of the energy cost field.

During the steady phase, the energy cost of a sensor node can be grouped to two categories, detection energy (E_d) used in detecting phenomena and transmission energy (E_t) consumed in transmitting data to other nodes. In an annulus, detection energy is in direct proportion to (k times, for example) its sensor nodes. Assume the number of total sensor nodes is N_{sensor}. The detection energy of annulus ceil(R/D) is

$$E_{d\_ceil(R/D)} = k \cdot N_{sensor} \cdot [\ F_s(R) - F_s(ceil(\frac{R}{d}) - 1)\ ] \qquad \text{Eqn. 4-1}$$

where ceil(x) is the ceiling function which rounds the elements of x to the nearest integer $\geq$x.

The transmitting energy of an annulus is in direct proportion to the data it has to relay. Because phenomena are assumed to be uniformly distributed, data is in direct proportion to the areas from which the data come. The transmitting energy of an annulus is in direct proportion to (x times, for example) areas whose data will be relayed by the annulus toward the origin. The transmitting detection energy of annulus ceil(R/D) is

$$E_{t\_ceil(R/D)} = x \cdot \pi \cdot [R^2 - (ceil(\frac{R}{D}) - 1)^2 \cdot D^2]. \qquad \text{Eqn. 4-2}$$

The total energy cost of annulus ceil(R/D) is

$$\begin{aligned} E_{ceil(R/d)} &= E_{d\_ceil(R/D)} + E_{t\_ceil(R/D)} \\ &= k \cdot N_{sensor} \cdot [\, F_s(R) - F_s(ceil(\frac{R}{d}) - 1)\,] + x \cdot \pi \cdot [(R^2 - (ceil(\frac{R}{D}) - 1)^2 \cdot D^2]. \end{aligned}$$

Eqn. 4-3

The transmitting energy of annulus ceil(R/D)-1 is directly proportional to sum of the areas of annulus ceil(R/D) and annulus ceil(R/D)-1, because it has to relay all data of both annuli to annulus ceil(R/D)-2. The energy cost of annulus ceil(R/D)-1 is

$$\begin{aligned} E_{ceil(R/D)-1} &= E_{d\_ceil(R/D)-1} + E_{t\_ceil(R/D)-1} \\ &= k \cdot N_{sensor} \cdot [F_s(ceil(\frac{R}{d}) - 1) - F_s(ceil(\frac{R}{d}) - 2)] + x \cdot \pi \cdot [R^2 - (ceil(\frac{R}{D}) - 2)^2 \cdot D^2]. \end{aligned}$$

Eqn. 4-4

Following the same approach, we can conclude that the energy cost of annulus i is

$$\begin{aligned} E_i &= E_{d\_i} + E_{t\_i}. \\ &= k \cdot N_{sensor} \cdot [F_s(i) - F_s(i-1)] + x \cdot \pi \cdot [R^2 - (i-1)^2 \cdot D^2]. \end{aligned} \qquad \text{Eqn. 4-5}$$

The total energy cost of all annuli is

$$E_{total} = \sum_{i=1}^{ceil(\frac{R}{D})-1} E_i + E_{ceil(R/D)}$$

$$= k \cdot N_{sensor} \sum_{i=1}^{ceil(\frac{R}{D})-1} [(F_s(i) - F_s(i-1)] + x \cdot \pi[(ceil(\frac{R}{D}) - 1) \cdot R^2 + D^2 \cdot \sum_{i=1}^{ceil(\frac{R}{D})-2} i^2] + E_{ceil(R/D)}$$

$$= k \cdot N_{sensor} + x \cdot \pi[ceil(\frac{R}{D}) \cdot R^2 + D^2 \frac{1}{6} ceil(\frac{R}{D}) \cdot (ceil(\frac{R}{D}) - 1) \cdot (2ceil(\frac{R}{D}) - 1)].$$

Eqn. 4-6

The energy density of annulus i is :

$$f_{e\_i} = \frac{E_i}{E_{total}}.$$

Eqn. 4-7

Because $f_{e\_i} = f_{s\_i}$,

$$f_{s\_i} = F_s(i) - F_s(i-1) = f_{e\_i}$$

$$= \frac{k \cdot N_{sensor} \cdot [F_s(i) - F_s(i-1)] + x \cdot \pi \cdot [R^2 - (i-1)^2 \cdot D^2]}{k \cdot N_{umber} + x \cdot \pi[ceil(\frac{R}{D}) \cdot R^2 + D^2 \frac{1}{6} ceil(\frac{R}{D}) \cdot (ceil(\frac{R}{D}) - 1) \cdot (2ceil(\frac{R}{D}) - 1)]}.$$

Eqn. 4-8

We get the energy cost density function, which is also the sensor node density function

$$f_{s\_i} = f_{e\_i} = \frac{R^2 - (i-1)^2 \cdot D^2}{ceil(\frac{R}{D}) \cdot R^2 + D^2 \frac{1}{6} ceil(\frac{R}{D}) \cdot (ceil(\frac{R}{D}) - 1) \cdot (2ceil(\frac{R}{D}) - 1)}$$

where $i = 1, 2, \cdots\cdots ceil(R/D)$ Eqn. 4-9

The energy cost density is a function of cluster diameter D, field radius R, and number of hops from the origin. Assume R=100, and D=10. The energy cost density function is shown as figure 4-3.

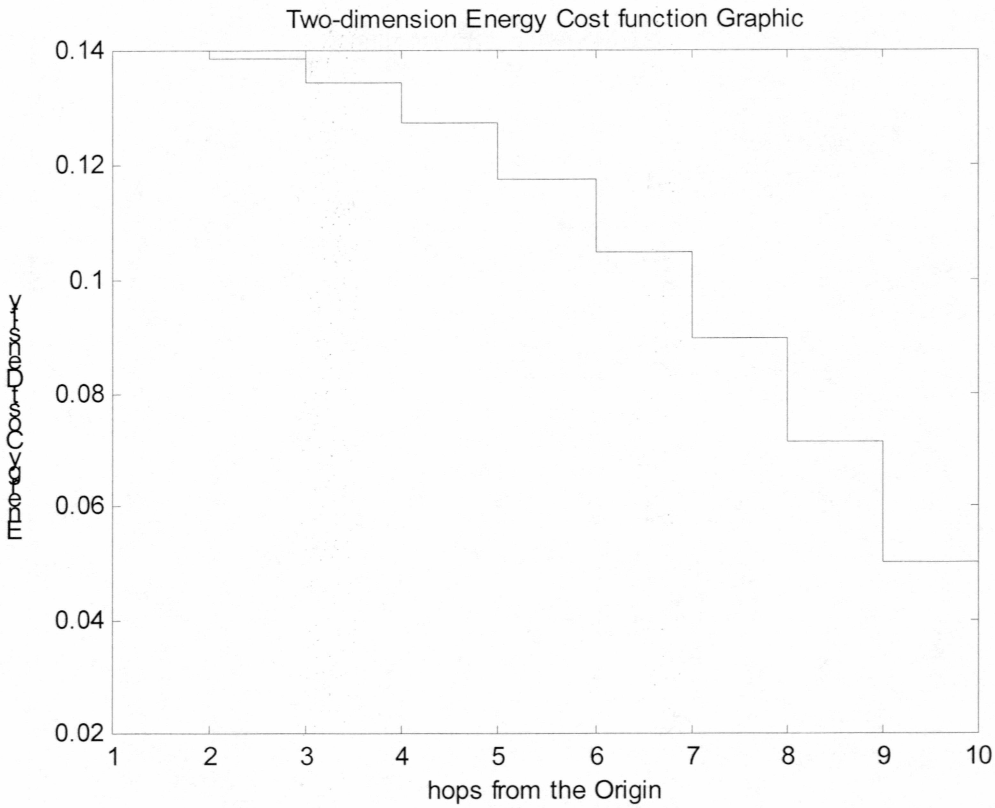

Figure 4-3. Energy cost density graphic.

Figure 4-4 shows the energy cost field in a 3-D graph. Appendix 2 is the matlab program used to visualize the energy cost density function .

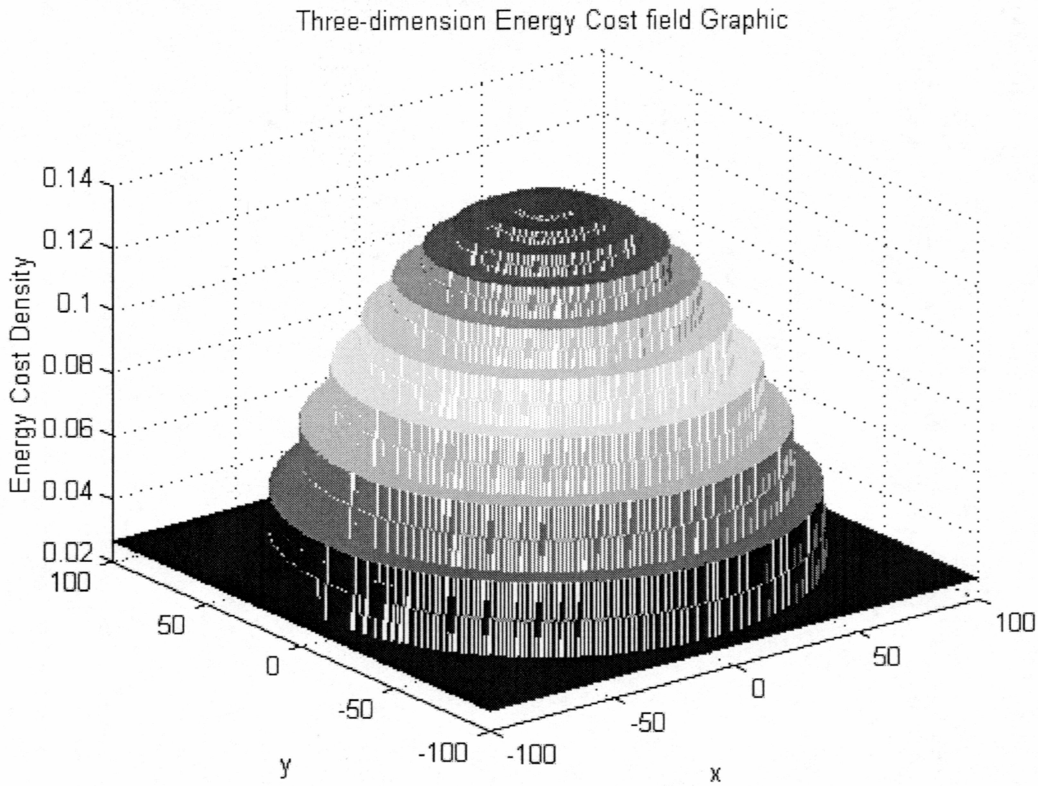

Figure 4-4. 3-D energy cost density graphic.

If we deploy sensor nodes per this density function, we can get optimized deployment: sensor nodes will use up their energy at almost the same time. Figure 4-5 is a simulation result of deployment according to the energy cost density function. The matlab program is listed in appendix 3.

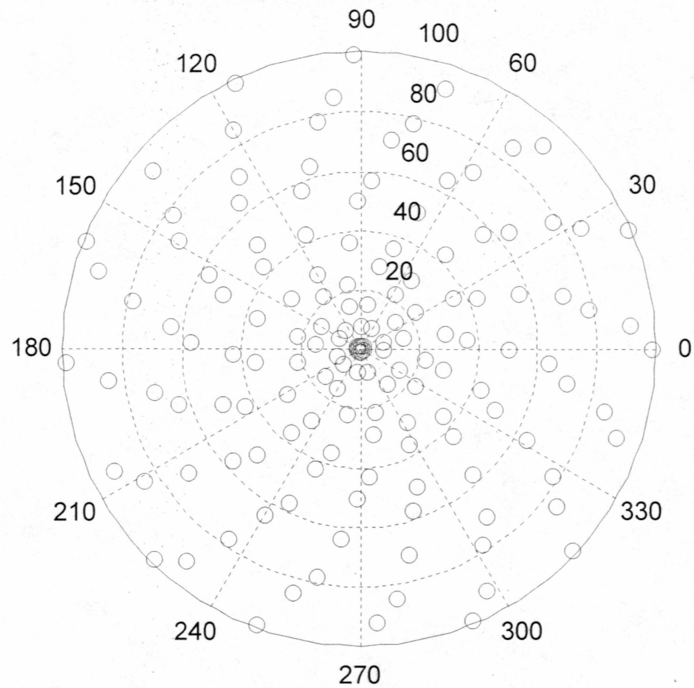

Figure 4-5. Simulation of optimum sensor nodes (150) deployment.

B. Optimum cluster size

In the last section, we discussed the energy cost in the energy cost field fashion. In this section, we will discuss the energy cost in a more detailed way--bit flow methods.

1. Radio link budget [11]

From basic transmission theory, we know that in a wireless link, the power received is determined by the transmitter power, transmitter antenna gain, and receiver antenna gain and path loss. The exact relation is given in Eqn. 4-10,

$$P_r(d) = \frac{P_t G_t G_r}{L_p}$$

Eqn. 4-10

where

P_r is received power,

P_t is transmitter power,

G_t is transmitter antenna gain,

G_r is receiver antenna gain, and

L_p is path loss.

Path loss accounts for the way energy spreads out as an electromagnetic wave travels away from a transmitting source. From chapter 2, we know that path loss is different in different wave propagation environments. Path loss is a function of path distance, path loss factor and so on. In the free space model, path loss is given in Eqn. 4-11,

$$L_{p\_freespace} = \left(\frac{4\pi}{\lambda}\right)^2 d^2 \qquad \text{Eqn. 4-11}$$

where

d is path distance and

λ is wavelength.

In the two-ray model, path loss is given in Eqn. 4-12,

$$L_{p\_tworay} = \frac{1}{h_t^2 h_r^2} d^4 \qquad \text{Eqn. 4-12}$$

where

d is path distance,

h_t is height of the transmitting antenna above ground, and

h_r is height of the receiving antenna above ground.

In the shadowing model, from Eqn. 2-6, we have

$$\frac{\overline{P_r(d)}}{P_r(d_0)} = \left(\frac{d}{d_0}\right)^{-\beta} 10^{\frac{X_{dB}}{10}}. \qquad \text{Eqn. 4-13}$$

From Eqn. 4-13, we get

$$\overline{P_r(d)} = P_r(d_0)\left(\frac{d}{d_0}\right)^{-\beta} 10^{\frac{X_{dB}}{10}} = \frac{P_t G_t P_r}{\left(\frac{4\pi d_0}{\lambda}\right)^2}\left(\frac{d}{d_0}\right)^{-\beta} 10^{\frac{X_{dB}}{10}}.$$

$$\overline{P_r(d)} = \frac{P_t G_t P_r}{\left(\frac{4\pi d_0}{\lambda}\right)^2}\left(\frac{1}{d_0}\right)^{-\beta} 10^{\frac{X_{dB}}{10}} d^{-\beta}. \qquad \text{Eqn. 4-14}$$

From Eqn. 4-14, we get path loss in the shadowing model as

$$L_{p\_shadowing} = \left(\frac{4\pi d_0}{\lambda}\right)^2 (d_0)^{-\beta} 10^{\frac{-X_{dB}}{10}} d^{\beta}. \qquad \text{Eqn. 4-15}$$

Now we can summarize equation 4-11, 4-12 and 4-15 to

$$L_p = \beta_n d^n. \qquad \text{Eqn. 4-16}$$

where,

n is path loss factor, for free space model, n=2, for two-ray model, n=4,

d is path distance and

β_n is path loss constant if path loss factor is fixed. When n=2, $\beta_2 = \left(\frac{4\pi}{\lambda}\right)^2$.

Substituting Eqn. 4-16 into Eqn. 4-10 yields Eqn. 4-17,

$$P_t = \frac{P_r}{G_t G_r} \beta_n d^n . \qquad \text{Eqn. 4-17}$$

Power transmitted or received per bit equals to the energy over the data rate, as given in Eqn. 4-18 and Eqn. 4-19

$$Eb_r = \frac{P_r}{R}, \qquad \text{Eqn. 4-18}$$

$$Eb_t = \frac{P_t}{R}, \qquad \text{Eqn. 4-19}$$

where,

Eb_r is Energy received per bit,

Eb_t is Energy transmitted per bit;

P_r is power received at receiver,

P_t is power transmitter power, and

R is data rate.

Substituting Eqn. 4-18 and Eqn. 4-19 into Eqn. 4-7 yields Eqn. 4-20

$$Eb_t = \frac{Eb_r}{G_t G_r} \beta_n d^n . \qquad \text{Eqn. 4-20}$$

To receive a message successfully, received power must be above the threshold, which is determined by the type of modulation scheme, type of detection used and the amount of noise present. The exact equation is given in

$$Eb_r = \left(\frac{S}{N}\right)_b kT_{sys} \qquad \text{Eqn. 4-21}$$

where,

$\left(\frac{S}{N}\right)_b$ is required SNR (Signal to Noise Ratio) per bit,

k is Boltzman's constant, and

T_{sys} is system temperature.

Substituting Eqn. 4-21 into Eqn. 4-20 we get Eqn. 4-22

$$Eb_t = \left(\frac{S}{N}\right)_b \frac{kT_{sys}}{G_t G_r} \beta_n d^n$$

$$= K\beta_n d^n . \qquad \text{Eqn. 4-22}$$

where, $K = \left(\frac{S}{N}\right)_b \frac{kT_{sys}}{G_t G_r}$.

In order to make a successful transmission, the transmission power should be greater than Eb_t, which is a function of path distance, path loss factor, path loss constant, required Signal to Noise Ratio per bit, transmitter antenna gain, receiver antenna gain, and system temperature, as described in Eqn. 4-22.

2. Transport bit energy [22] - [24]

When a sensor node has data to send out, it sends its data to the cluster head first. After aggregating data, the cluster head sends the data to the up stream neighbor cluster, which will forward the data toward the base station. We can calculate the intra-cluster and inter-cluster energy dissipation respectively.

Energy dissipated for a bit in a cluster consists of computational energy and communicational energy. Computional energy includes energy used in the process of getting the data and energy used in aggregating data by the cluster head. Communicational energy is the energy dissipated in transmitting data to the cluster head.

$$E_{\text{intra\_cluster}} = E_{\text{intra\_comp}} + E_{\text{int}\,ra\_trans}$$

$$= \mathrm{E}_{\text{intra\_comp}} + K\beta_{n1} d_{\text{intra\_cluster}}{}^{n1}. \qquad \text{Eqn. 4-23}$$

Energy dissipated between clusters consists of computational and communication energy too. Because the distance between clusters may be many times farther than the distance between intra-cluster nodes, we may need to use different models for calculating inter-cluster transmission energy. Assuming the data has to pass N hops before arriving at the base station, the energy dissipated in transmitting per bit in multi-hop is given in

$$E_{\text{inter\_cluster}} = N \cdot E_{\text{inter\_comp}} + N \cdot E_{\text{inter\_trans}} = N \cdot E_{\text{inter\_comp}} + N \cdot Eb_{\text{inter\_trans}}.$$

$$E_{\text{inter\_cluster}} = N \cdot E_{\text{inter\_comp}} + N \cdot K\beta_{n2} d_{\text{inter\_cluster}}{}^{n2}. \qquad \text{Eqn. 4-24}$$

Now, assuming the aggregation method compress data with a ratio of L:1. That means when the cluster head receives L bits from its members, it will send out only 1 bit to the base station. The total energy consumed in transmitting a bit from a sensor node to the base station should be

$$Eb = E_{\text{intra\_cluster}} + E_{\text{inter\_cluster}}.$$

$$= \mathrm{E}_{\text{intra\_comp}} + K\beta_{n1} d_{\text{intra\_cluster}}{}^{n1} + \frac{1}{L} N \cdot E_{\text{inter\_comp}} + N\frac{1}{L} \cdot K\beta_{n2} d_{\text{inter\_cluster}}{}^{n2}. \qquad \text{Eqn. 4-25}$$

3. Optimization of cluster size

It is very complicated to calculate the effect of cluster size on energy consumed. Here, I analyze this based upon some assumptions and simplifications.

Assume sensor clusters are distributed uniformly in the investigating area. As discussed in the previous section, a cluster near the origin may have more sensor nodes than a farther cluster has. Assuming the distance from a cluster to the origin is r, as shown in figure 4-6, the probability density function of clusters is

$$f_r(r) = \begin{cases} \dfrac{2r}{R^2} & r \in [0, R] \\ 0 & others \end{cases} . \qquad \text{Eqn. 4-26}$$

The respective distribution function is

$$F_r(r) = \begin{cases} \dfrac{r^2}{R^2} & r \in [0, R] \\ 0 & others \end{cases} . \qquad \text{Eqn. 4-27}$$

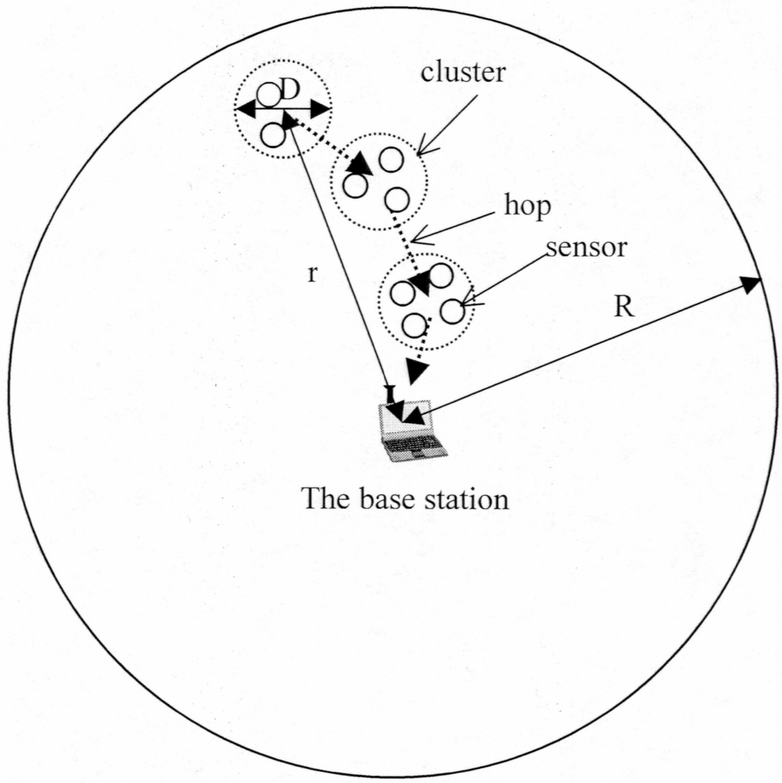

Figure 4-6. Calculation of the bit energy cost.

Assuming data from a cluster must be forwarded n hops before it arrives the origin, we get the hop density function,

$$P\{N = n) = P\{\ (n-1)D < n < nD\ \} = F_r(nD) - F_r(nd-d)$$

$$= \begin{cases} \dfrac{D^2}{R^2}(2n-1) & n = 1,2,\ldots ceil(\dfrac{R}{D}) - 1 \\ \dfrac{R^2 - ((n-1)\cdot D)^2}{R^2} & n = ceil(\dfrac{R}{D}) \end{cases} \quad . \qquad \text{Eqn. 4-28}$$

Assuming D=10 m and R=100 m, the calculated hop density function is shown in figure 4-7.

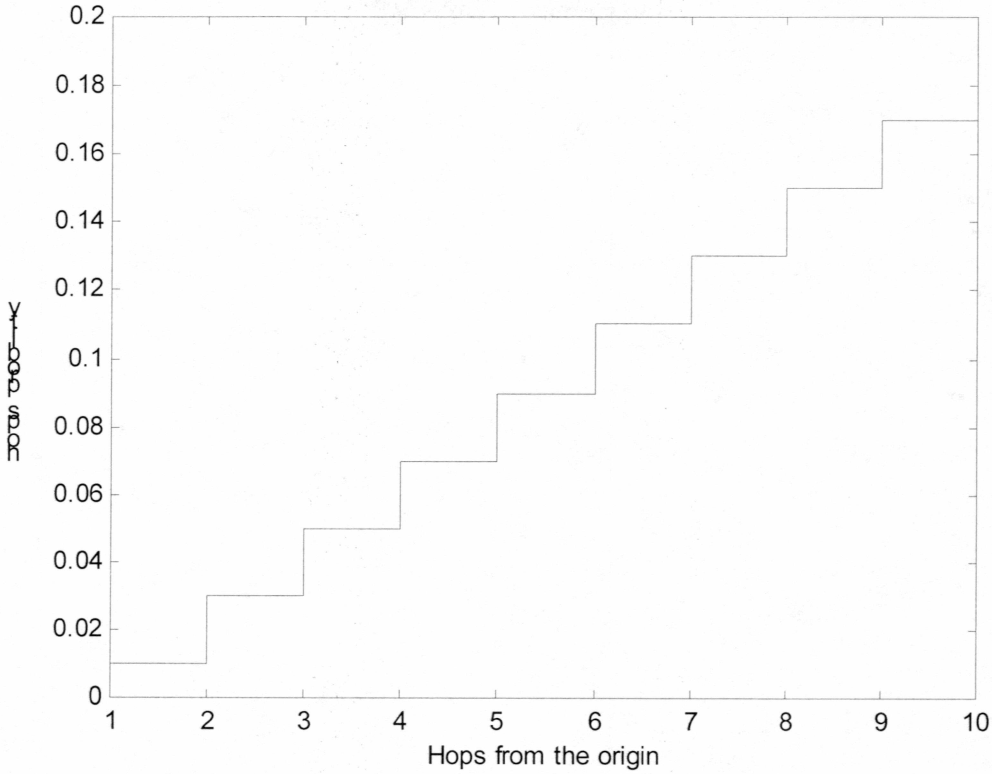

Figure 4-7. Hop density function.

We assume each node in the network has the same amount of data to send, and furthermore, we assume each cluster contains u node members. From Eqn. 4-16 and Eqn. 4-17, we can get expected Eb,

$$E(Eb) = \sum_{n=1}^{ceil(\frac{R}{D})} Eb \cdot P(N = n) .$$

$$E(Eb) = \sum_{N=1}^{ceil(\frac{R}{D})-1} [\,\mathrm{E}_{\text{intra\_comp}} + uK\beta_{n1}D^{n1} + Nu/L \cdot E_{\text{inter\_comp}} + Nu/L \cdot K\beta_{n1}(3D)^{n2}][\frac{D^2}{R^2}(2N-1)]$$

$$+[\mathrm{E}_{\text{intra\_comp}} + uK\beta_{n1}D^{n1} + Nu/L \cdot E_{\text{inter\_comp}} + Nu/L \cdot K\beta_{n2}(3D)^{n2}]\frac{R^2 - ((ceil(R/D)-1)\cdot D)^2}{R^2}.$$

Eqn. 4-29

For example, assume $(S/N)_b$=10, k=1.38×10^{-23} [J/K], R=100 [m], $T_{sys}=T_{background}+T_{trans}(F-1)$=290+290(3-1)=870 [K], λ=0.15 [m], f=2 [GHz], G_t=1, G_r=1, $E_{inter\_comp} = 50\times10^{-12}\ [J]$, u=20, and L=5. From Eqn. 4-18, Eb is a function of D. The relation between D and Eb is shown in figure 4-8.

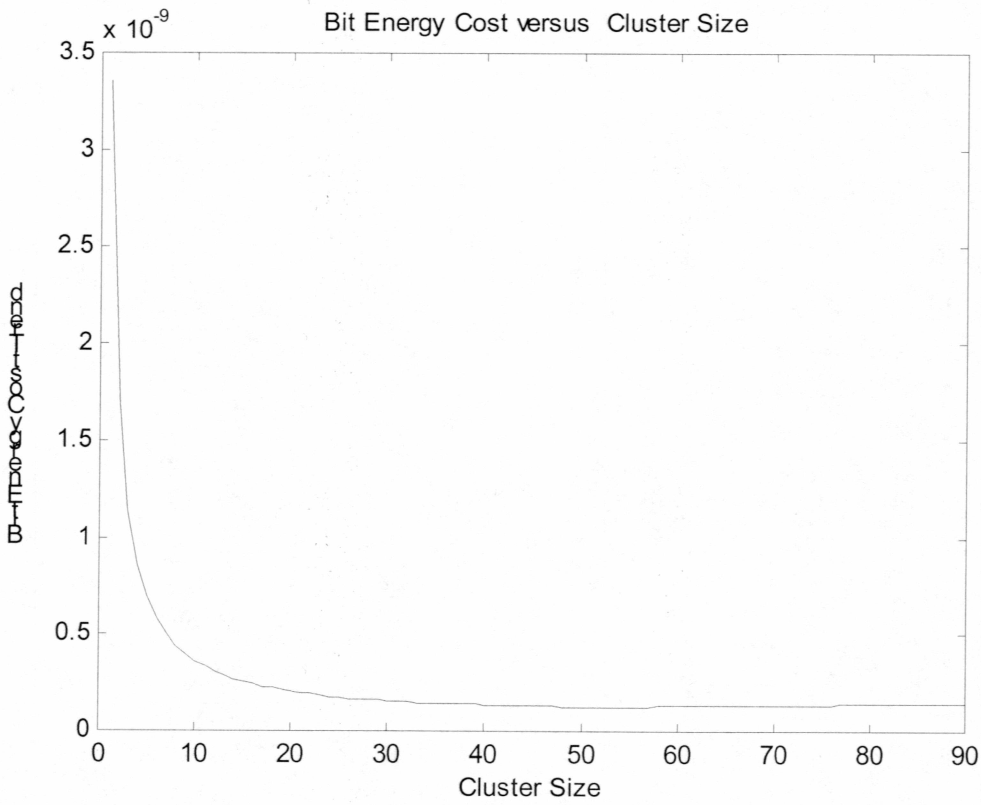

Figure 4-8. Energy cost vs. cluster size ($P_r \propto 1/d^2$).

From figure 4-8, we can see that if cluster size is smaller than 20 m, the transport bit energy will increase sharply. The optimum cluster size can be deduced from Eqn. 4-29. Optimum cluster size should make minimization energy dissipation in transmitting a bit from a sensor node to the base station. That is

$$\frac{d(E(Eb))}{d(D)}=0,$$

$$\frac{d\{\sum_{N=1}^{[\frac{R}{D}]}[\mathrm{E}_{\text{intra\_comp}}+uK\beta_{n1}D^{n1}+\frac{Nu}{L}E_{\text{inter\_comp}}+\frac{Nu}{L}K\beta_{n2}(3D)^{n2}][\frac{D^2}{R^2}(2N-1)]\}}{d(D)}=0. \text{ Eqn.4-30}$$

We can get optimum cluster size D by solving this equation.. After getting the optimum cluster size, we can design the transmitting power with which the network will cluster in the optimum cluster size.

C. Propagation condition

The propagation condition is another important parameter in energy cost. In the ideal condition, the received power has an inverse squared relationship with transmitter-receiver separation distance, $P_r \propto 1/d^2$. In worse conditions, the received power may have an inverse cubic relationship with distance d, $P_r \propto 1/d^3$ or even worse. If the distance between the transmitter and receiver is small enough, the propagation condition can be regarded as ideal condition; as the distance increases, the propagation condition may become worse. In the nanosensor networks described in this paper, the intra-cluster propagation and inter-cluster propagation have different transmission distances. Different propagation models may be used for inter-cluster and inter-cluster propagation.

The worse the propagation condition, the more energy will be consumed. Optimum cluster size will change as the condition changes. When the condition becomes worse, the cluster size becomes smaller. This is reasonable. When radio wave decays sharply, long distance transfer will consume much more energy than short distance transfer. Figures 4-9 and 4-10 show the bit energy cost with different inter-cluster path loss exponents from $d^{-1.5}$ to d^{-4}, and from d^{-5} to d^{-7} respectively, while the intra-cluster path loss exponent is d^{-2}. The bit energy cost with n2 = -2 and n2 = -3 is much less then that with n2 = -4. In order to show all curves in the same figure, the big energy cost with n2 = -2 and n2 = -3 are multiplied by 2000 and 100 respectively. The implemented programs are listed in appendix 4 and appendix 5.

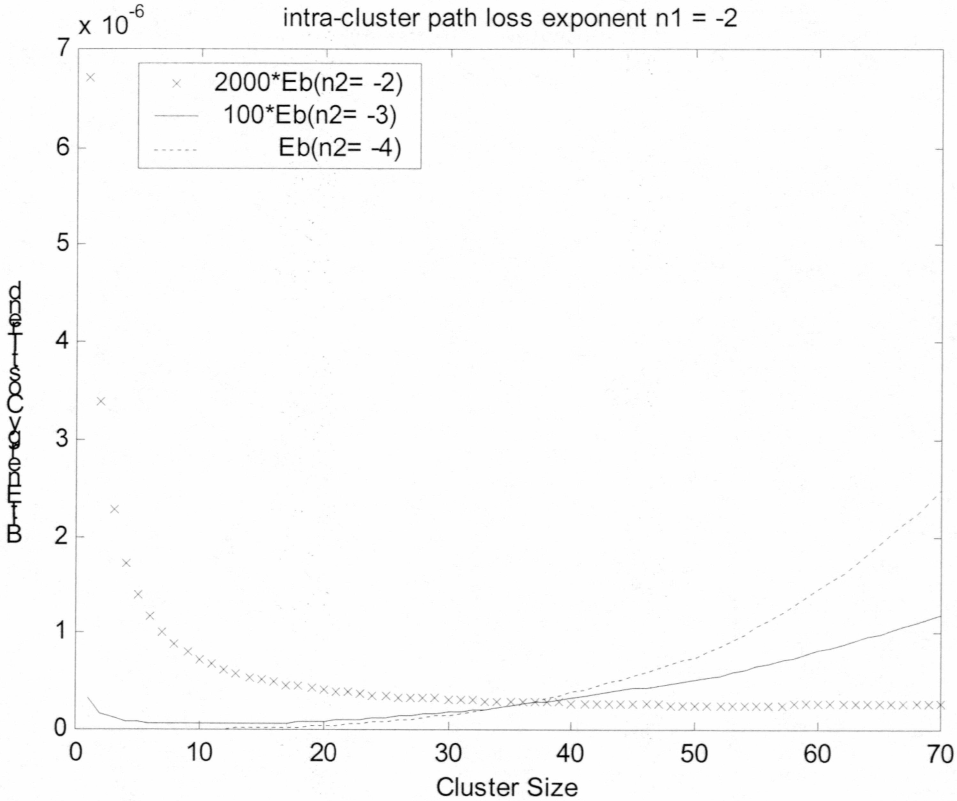

Figure 4-9. Propagation condition effect in bit energy cost (1).

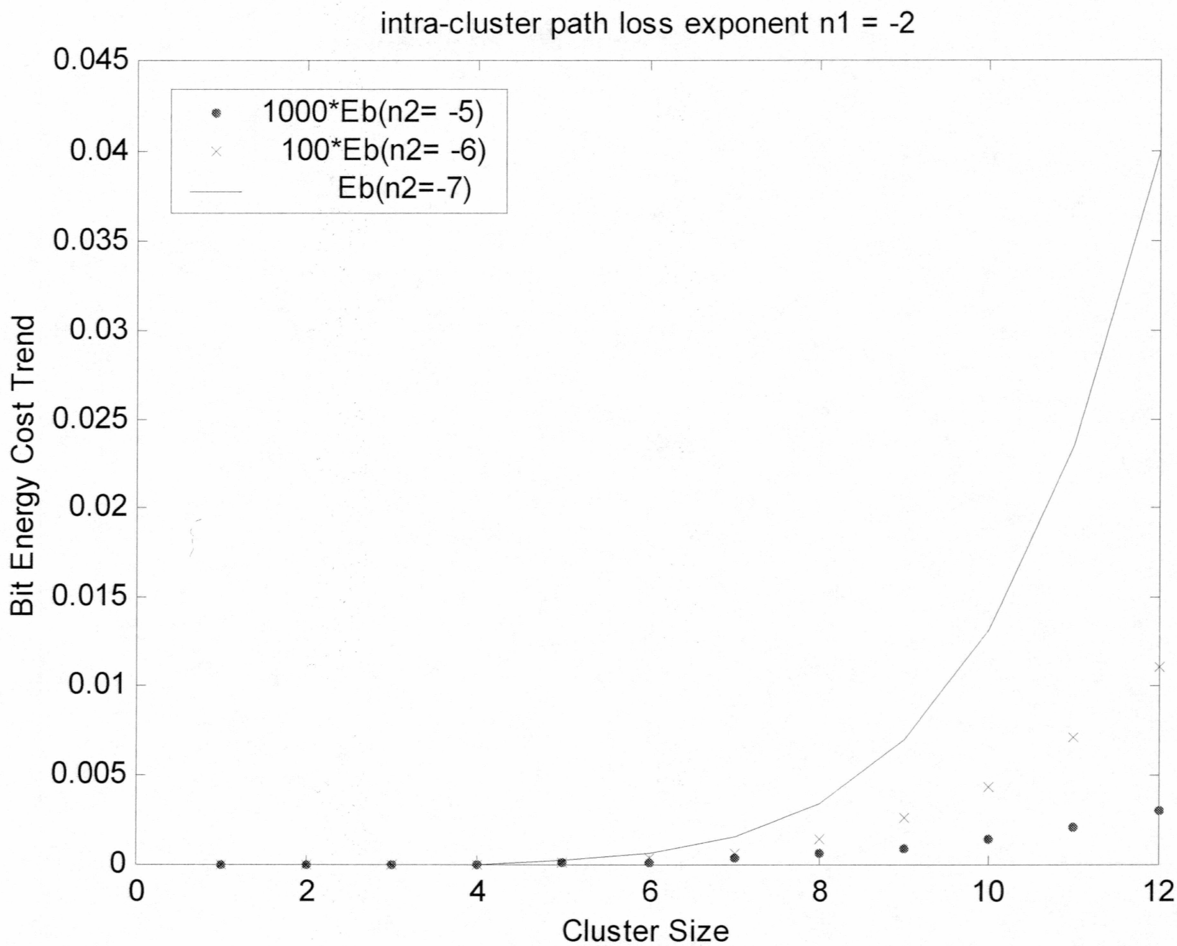

Figure 4-10. Propagation condition effect in bit energy cost (2).

From figure 4-9 and 4-10, we can see energy cost increases sharply when the propagation condition gets worse. The optimum cluster size decreases when the propagation condition becomes worse.

Figure 4-11 shows the bit energy cost versus cluster sizes with different inter-cluster propagation conditions, while intra-cluster propagation condition is $P{\sim}d^{-3}$. Figure 4-12 shows the bit energy cost versus cluster sizes with different inter-cluster propagation conditions, while intra-cluster propagation condition is P~d^{-4}.

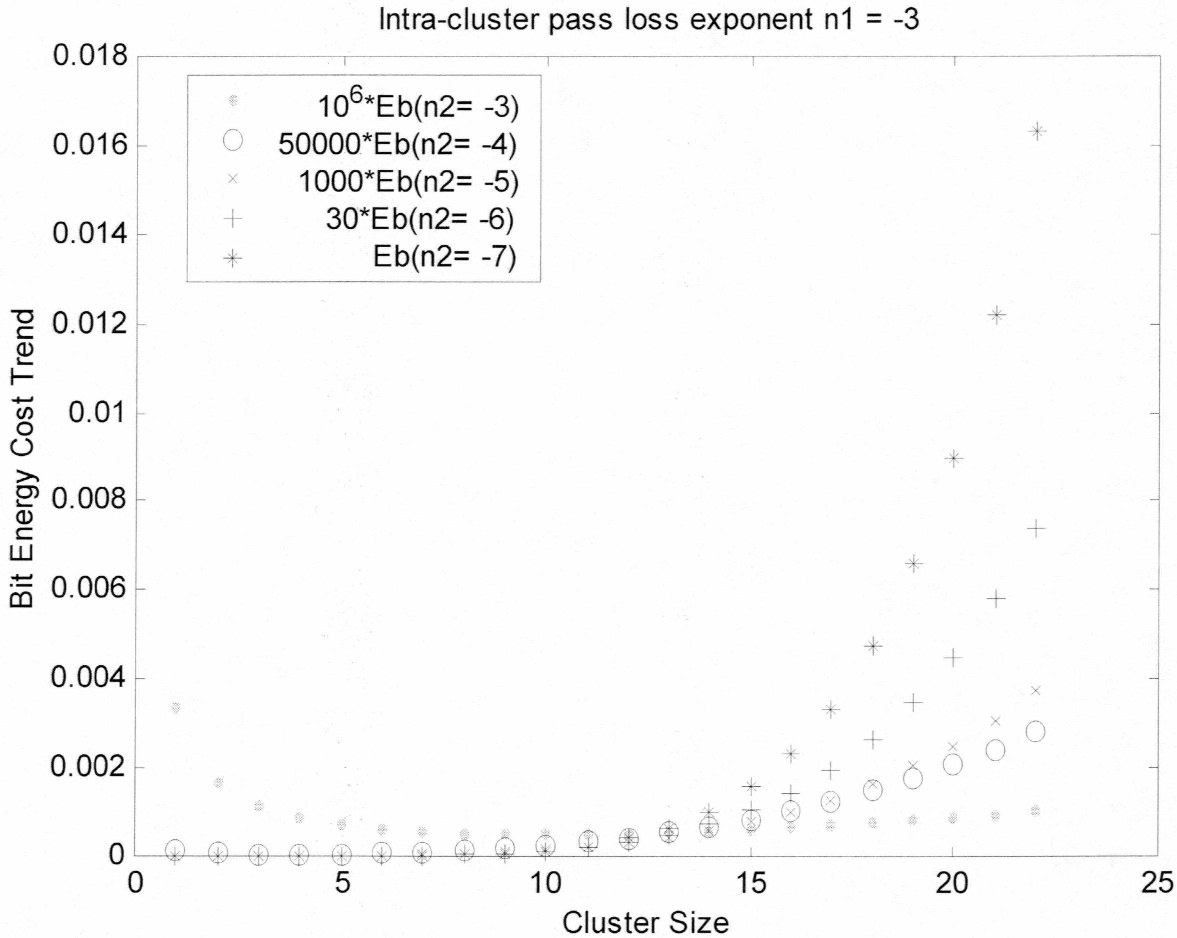

Figure 4-11. Propagation condition effect in bit energy cost (3).

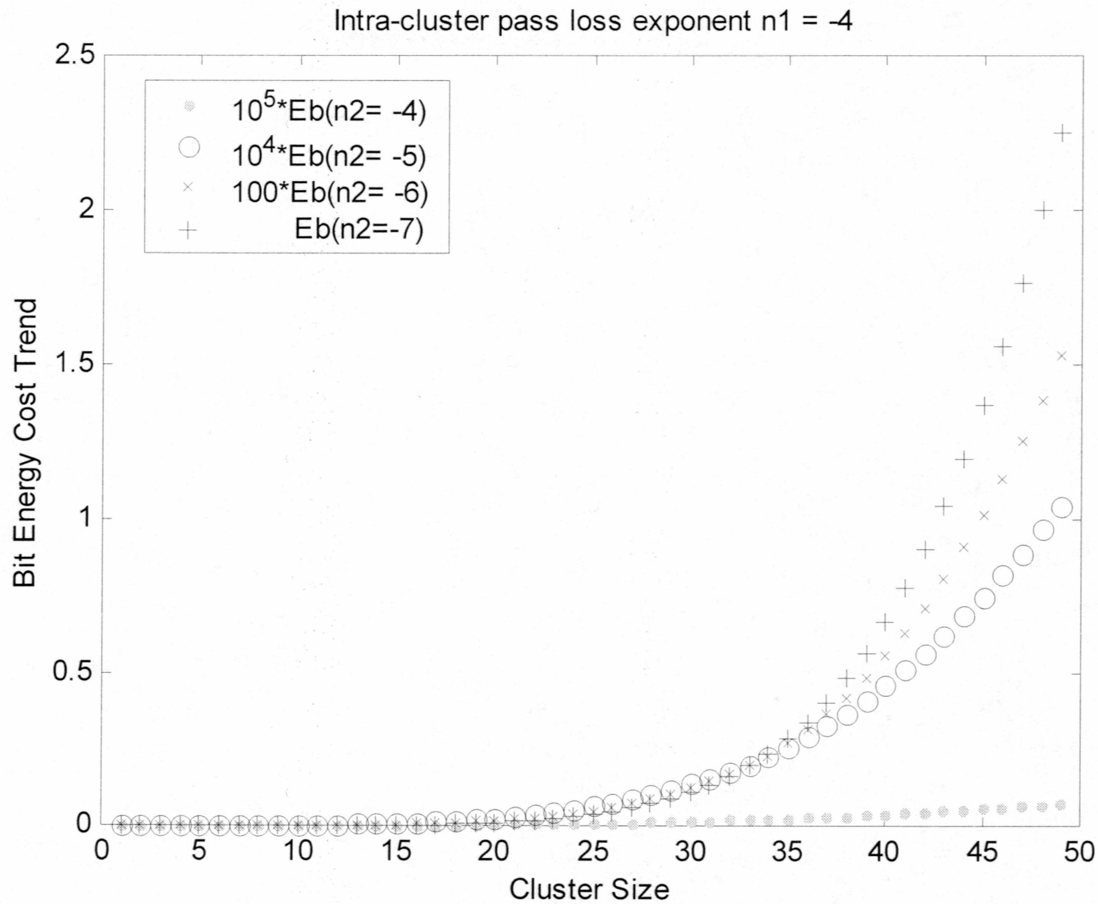

Figure 4-12. Propagation condition effect in bit energy cost (4).

D. Data compression ratio

Data compression ratio is also a key element in determining energy cost. Energy cost has a inverse relationship with the data compression ratio, as shown in figure 4-13. This is obvious. The bigger the compression ration, the smaller the number of bits that are needed to be sent to the base station.

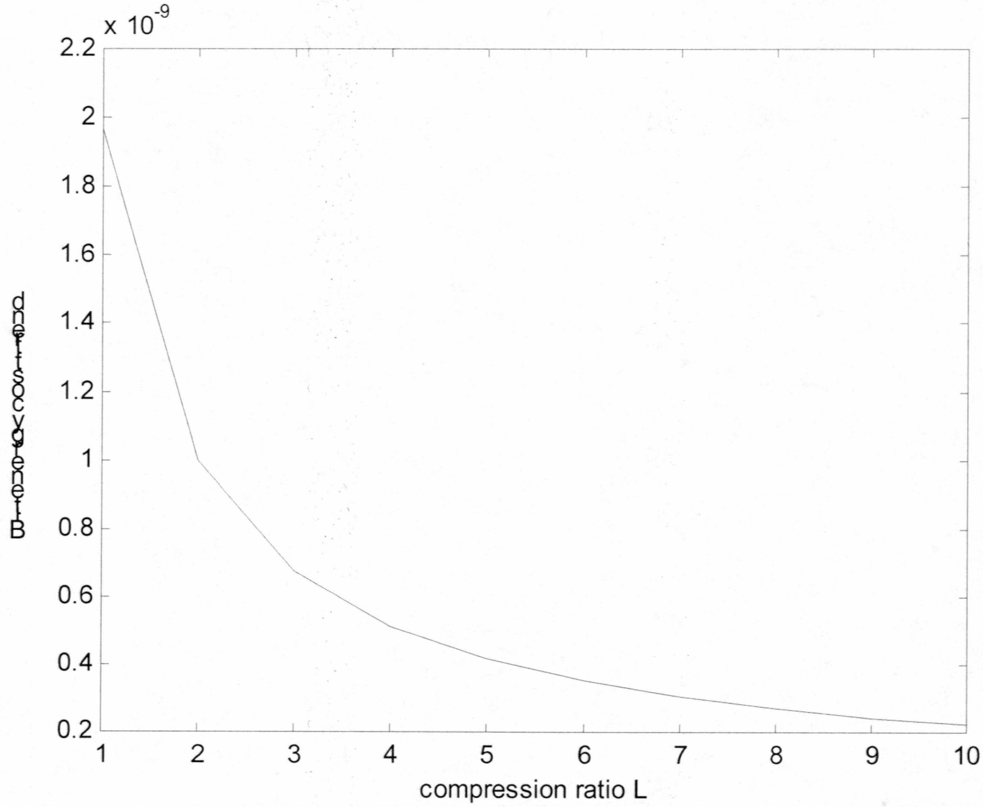

Figure 4-13. Bit energy cost vs. compression ratio.

The remaining question is when will a big compression ratio occur? How much can a cluster head aggregate data? When data correlation is low, is it still valuable to aggregate data?

1. Data correlation

Sensor nodes can only detect the source signal in a certain range. If a source signal is too far away from a sensor node, a sensor node can not detect the source because of source signal attenuation. Assuming sensors are omnidirectional, a senor node's detection range is a circle of radius r (maximum detection distance). Data correlation happens when detection range of sensor nodes overlaps. The probability of data correlation can be defined as

$$p_{correlation} = \frac{area\ overlapped}{total\ area\ seen\ by\ at\ least\ one\ sensor}.$$

Let's investigate the most simple case--two sensors in a scene. If two sensor nodes are almost at the same place, their detection ranges overlap almost totally, so the probability of data correlation tends to be 100%. The probability will decrease when two sensor nodes are separated far away from each other. When the distance between the two sensor nodes d equals to or is greater than two times of sensor nodes' maximum detection distance, 2*r, their detection ranges do not overlap at all, so the probability of data correlation is 0%. So the necessity of data correlation is that the distance between sensor nodes should less than two times maximum detection distance, d<2*r.

If there are many sensor nodes, the maximum probability of data correlation tends to 100%. What's minimum probability? Minimum probability happens when all sensors are at the circumference of the cluster boundary and number of sensor nodes tends to infinity, as shown in figure 4-14. The grey area is overlap area.

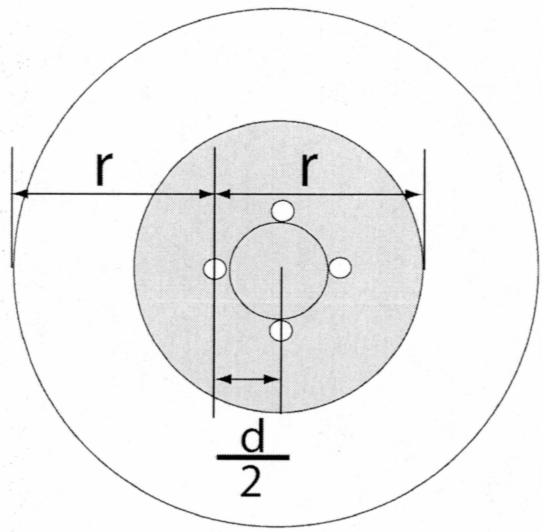

Figure 4-14. Minimum data correlation.

$$p_{correlation\_\min imum} = \frac{area\ overlapped}{total\ area\ seen\ by\ at\ least\ one\ sensor}$$

$$= \frac{\pi\left(r - \frac{d}{2}\right)^2}{\pi\left(r + \frac{d}{2}\right)^2} = \left(\frac{2r-d}{2r+d}\right)^2. \qquad \text{Eqn. 4-31}$$

2. Data aggregation

When the probability of data correlation is too small, the compression ratio is small. It is not valuable to aggregate data, because the aggregating data process consumes energy. If the energy deference between transmitting aggregated data and transmitting the original data is less than energy dissipated in data aggregating process, it is not valuable to aggregate data. Assume the aggregation method can compress the data with ratio of L:1. This means that for every L bits that must be sent to the base station when no data aggregation is performed, only 1 bit must be forwarded to the base station through multi-hop when local data aggregation is performed. The energy to perform local data aggregation and transmit the aggregated 1 bit is

$$E_{aggregate} = L \times E_{agg\_process} + Eb. \qquad \text{Eqn. 4-32}$$

Eb is described in Eqn. 4-28. The energy to transmit all L bits of data directly to base station through multi-hop is

$$E_{non\_aggregate} = L \times Eb. \qquad \text{Eqn. 4-33}$$

The energy used in the aggregation process is less than the energy consumed in sending all the unprocessed data to the base station when

$$E_{aggregate} < E_{non\_aggregate}.$$

$$L \times E_{agg\_process} + Eb < L \times Eb.$$

$$\frac{E_{agg\_process}}{Eb} < \frac{L-1}{L}. \qquad \text{Eqn. 4-34}$$

V. Conclusions and results

This paper discusses nanosensor network communication protocols and energy cost. In this protocol stack, sensor nodes cooperate by grouping into clusters and sending data through multi-hops in cluster scale. The clustering and routing algorithm forms clusters and sets up routing by using a distributed algorithm, where nodes make autonomous decisions without any centralized control. No long-distance communication with the base station is required. Sensor nodes within a cluster take turns to be the cluster head. A TDMA scheme decreases energy dissipation and interference. There is no set-up overhead at the beginning of each round. A cluster head need not remember the whole state of the sensor network. It only needs to remember its neighbor clusters. Cluster scale multi-hop is more stable than a single node. The clustering and routing scheme can preserve its structure when a few nodes are moving and the topology change slowly. The sensor network is scaleable under the protocol stack.

One of the operational challenges in nanosensor networks is its limited irreplaceable power resources. By analyzing energy, several interesting results are achieved. Optimum sensor node deployment is found to allow all sensor nodes to use up their energy at almost the same time. The optimum cluster size is estimated to get the maximum lifetime of the whole sensor networks. Propagation conditions are the key parameter of transmitting energy cost. Different propagation conditions are compared in energy cost.

References

[1] Malik Tubaishat and Sanjay Madria, "Sensor networks: an overview", IEEE Potentials, pp. 20-23, April/May, 2003.

[2] I.F.Akyildiz, W.Su, et al, "Wireless sensor networks: a survey" Computer Networks Journal, Vol. 38, pp. 393-422, No. 4, 2002.

[3] I.F.Akyildiz, W. Su, et al, "A Survey on Sensor Networks", IEEE Communication Magazine, Vol. 40, No. 8, pp. 102-114, August 2002.

[4] http://www.isi.edu/scadds/.

[5] http://www-mtl.mit.edu/research/icsystems/uamps/.

[6] http://bwrc.eecs.berkeley.edu/Research/Pico_Radio/Default.htm.

[7] http://robotics.eecs.berkeley.edu/~pister/SmartDust/.

[8] http://www.janet.ucla.edu/WINS/.

[9] Ramakrishna Janaswamy, "Radiowave Propagation and Smart Antennas for Wireless Communications" Kluwer Academic Publishers, 2001.

[10] P.Mohana Shankar, "Introduction to Wireless Systems". Johin Wiley & Sons, INC. 2002.

[11].Theodore S. Rappaport, "Wireless Communications Principles and Practice", second edition. Prentice-Hall, 2002.

[12] Ray A. Foran, Thad B. Welch, et al., "Very near ground radio frequency propagation measurements and analysis for military applications", MILCOM, Nov. 1999.

[13] William stallings, "Data & Computer Communication", sixth edition, Prentice Hall. 1999.

[14] Suman Banerjee, Samir Khuller, "A clustering scheme for hierarchical control in multi-hop wireless networks", Proceedings of IEEE INFOCOM, April 2001.

[15] Chunhung Richard Lin, Mario Gerla, "Adaptive clustering for mobile wireless networks", IEEE Journal in Communication, Vol. 15, pp. 1265-1275, No. 7, September 1997.

[16] John W. Harris, Horst Stocker, "Handbook of Mathematics and Computational Science", Springer, 1998.

[17] Katayoun Sohrabi, Jay Gao, et al, "Protocols for self-organization of a wireless sensor network", IEEE Personal Communications, Vol. 7, pp. 16-27, No. 5, October 2000.

[18] Raghuvel S. Bhuvaneswaran, Jacir L. Bordim, et al, "Fundamental protocols for wireless sensor networks", IEICE Trans. on Fundamentals, Vol. E85-A, pp. 2479-2488, No. 11, Nov., 2002.

[19] Weilian Su, Ian F. Akyildiz, "A Stream Enabled Routing (SER) Protocol for Sensor Networks", Proceedings of Med-hoc-Net 2002, Sardegna, Italy, September 2002.

[20] F. Ye, S. Lu, et al., "GRAdient broadcast: A robust, long-lived large sensor network", UCLA CS Technical Report, September 2001.

[21] Chieh-Yih Wan, Andrew T. Campbell, et al, "PSFQ, A reliable transport protocol for wireless sensor networks", First ACM International Workshop on Wireless Sensor Networks and Application (WSNA), 2002.

[22] Danny D.Patel, "Energy in Ad-Hoc Networking for the PicoRadio", University of California at Berkeley, Master degree thesis, 2000.

[23] Stephanie Lindsey, Cauligi Raghavendra, et al., "Data gathering algorithms in sensor networks using energy metrics" IEEE Transactions on Parallel and Distributed Systems, Vol. 13, pp. 924-935, No. 9, September 2002.

[24] Wendi Beth Heinzelman, "Application-specific protocol architectures for wireless networks", Massachusetts Institute of Technology, Ph.D. thesis, June 2000.

Appendix 1

Uniform deployment of sensor nodes

```
% This program simulate uniform deployment of sensor nodes in the field.

clear all;
R=100;          % The radius of the whole detection circle (field).
Number=100;     % Total number of sensor nodes which will be deployed in
the field.
step=10;        % step of calculation.
r=[];theta=[]; % Cartesian polar coordinate
x=[];y=[];
I=1;
for rr=0:step:R
   for J=1:floor(Number/R^2*(2*step*rr-step^2))
      r(I)=rr;
      theta(I)=27*2*pi*I/Number; % to separate sensor node wide enough
      x(I)=r(I)*sin(theta(I)); % translate polar coordinate into
Cartesian coordinate
      y(I)=r(I)*cos(theta(I));
      I=I+1;
   end
end
polar(theta,r,'O');
%title('Simulation of Uniform Deployment of Sensor Nodes(150) in Polar
Coordinate')
figure
plot(x,y,'O');
%xlabel('x');
%ylabel('y');
%title('Simulation of Uniform Deployment of Sensor Nodes (100)in
Cartesian Coordinate');

clear all;
```

Appendix 2

Energy Cost Field Function

```
% This program generates graphics of energy cost field function: 2D, 3D,
and contour graphic.

clear all;
R=100;  % The radius of the whole detection circle(field).
D=10;   % The diameter of clusters
Maxihop=ceil(R/D)-1;
for I=1:1:Maxihop+1
   r(I)=I;
   energy(I)=EnergyCostFunction(R,D,I,Maxihop);
end
stairs(r,energy);  %Two-dimension Energy Cost Density Graphic
xlabel('hops from the Origin');
ylabel('Energy Cost Density');
title('Two-dimension Energy Cost function Graphic');

figure;
[X,Y] = meshgrid(-R:1:R, -R:1:R);
t=sqrt(X.^2+Y.^2);
for I=1:1:length(t)
   for J=1:1:length(t)
      if t(I,J)>R
         Z(I,J)=EnergyCostFunction(R,D,Maxihop+1,Maxihop);
      else
         Z(I,J)=EnergyCostFunction(R,D,ceil(t(I,J)/D),Maxihop);
      end
   end
end
mesh(X,Y,Z); %Three-dimension Energy Cost Density Graphic
xlabel('x');
ylabel('y');
zlabel('Energy Cost Density');
title('Three-dimension Energy Cost field Graphic');
figure;
contour(X,Y,Z); %Energy Cost Density Contour Graphic
xlabel('x');
ylabel('y');
title('Energy Cost Density Contour Graphic');
clear all;
function density=EnergyCostFunction2(R, D,I,Maxihop)
% This function calculate the energy cost density of annulus I.
% R:        The radius of the whole detection circle(field).
% D:        The diameter of clusters
% I:        The number of annulus.
% Maxihop:  The Maximum number of annulus.
total=Maxihop+1-D^2/R^2/6*Maxihop*(Maxihop+1)*(2*Maxihop+1);
density=(1-(D*(I-1))^2/R^2)/total; % Calculate the energy cost percent
of annulus I.
```

Appendix 3

Optimum deployment of sensor nodes

```
% This program simulate optimum deployment of sensor nodes in the field

clear all;
R=100;    % The radius of the whole detection circle(field).
D=8;     % Calculation step;
r=[];theta=[];  % Polar Coordinate
x=[];y=[];      % Cartesian coordinate
Number=150; % Total number of sensor nodes which will be deployed in
the field.
Maxihop=ceil(R/D)-1;
I=1;
for rr=0:D:R
   for J=1:floor(Number*EnergyCostFunction(R,D,ceil(rr/D),Maxihop))
      r(I)=rr;
      theta(I)=27*2*pi*I/Number;
      x(I)=r(I)*sin(theta(I)); % translate polar coordinate into
Cartesian  coordinate
      y(I)=r(I)*cos(theta(I));
      I=I+1;
   end
end
I
polar(theta,r,'o');
title('Simulation of Optimization Deployment of Sensor Nodes in the
Field in Polar Coordinate');

figure;
plot(x,y,'O');
xlabel('x');
ylabel('y');
title('Simulation of Optimization Deployment of Sensor Nodes in the
Field in Cartesian Coordinate');

clear all;
```

Appendix 4

Bit energy cost while inter-cluster pass loss exponent d = -1.5 to -4

```
% This program generate graphic of Bit Energy Cost versus Propagation
Condition and Cluster Size.
clear all;
R=100; % The radius of the whole detection circular field.
Eb=[;];
d=[1:1:70];     % Cluster Size
n1=2,n2=1.5
for I=1:length(d)
    maxi_hop=ceil(R/d(I))-1;
    tmp=0;
    for J=1:1:maxi_hop
        tmp=tmp+BitEnergyFunction(R,d(I),J,n1,n2);
    end
        tmp=tmp+BitEnergyFunction2(R,d(I),J+1,n1,n2);

    Eb(1,I)=tmp;
end



for n=2:1:4  % Path loss exponent
for I=1:length(d)
        maxi_hop=ceil(R/d(I))-1;
    tmp=0;
    for J=1:1:maxi_hop
        tmp=tmp+BitEnergyFunction(R,d(I),J,n1,n);
    end
        tmp=tmp+BitEnergyFunction2(R,d(I),J+1,n1,n);

    Eb(n,I)=tmp;

end
end

plot(d,Eb(1,:)*1000,'m.');
%plot(d,Eb(1,:),'y.');

hold on;
plot(d,Eb(2,:)*2000,'rx');
%plot(d,Eb(2,:),'m*');

hold on;
plot(d,Eb(3,:)*100,'g-');

hold on;
plot(d,Eb(4,:),'b:');

%plot(d,Eb(3,:),'co');
```

```
legend('1000*P~d^-^1^.^5','2000*P~d^-^2',' 100*P~d^-^3','         P~d^-
^4');
xlabel('Cluster Size');
ylabel('Bit Energy Cost Trend');
title('Inter-cluster pass loss exponent d = -2');

hold off
clear all;
```

Appendix 5

Bit energy cost while inter-cluster pass loss exponent d = -5 to -7

```
% This program generate graphic of Bit Energy Cost versus Propagation
Condition and Cluster Size.
clear all;
R=100; % The radius of the whole detection circle(field).
Eb=[;];
d=[1:1:12];     % Cluster Size
n1=2;
for n=5:1:7  % Propagation Conditions
for I=1:length(d)
   maxi_hop=ceil(R/d(I))-1;
   tmp=0;
   for J=1:1:maxi_hop
      tmp=tmp+BitEnergyFunction(R,d(I),J,n1,n);
   end
      tmp=tmp+BitEnergyFunction2(R,d(I),J+1,n1,n);


   Eb(n-4,I)=tmp;

end
end

plot(d,Eb(1,:)*10000,'m.');
%plot(d,Eb(1,:),'y.');

hold on;
plot(d,Eb(2,:)*1000,'rx');
%plot(d,Eb(2,:),'m*');

hold on;
plot(d,Eb(3,:)*100,'g-');



legend('1000*P~d^-^5','  100*P~d^-^6','         P~d^-^7');
xlabel('Cluster Size');
ylabel('Bit Energy Cost Trend');
```

```
title('Inter-cluster pass loss exponent d = -2');


hold off
clear all;


function BitEnergy=BitEnergyFunction(R,D,J,n1,n2)
% This function calculate bit energy cost given cluster size, number of
hops, and propagation
% R, The radius of the whole detection circle(field).
% D, cluster size;
% J, number of hops from the origin;
% n1, propagation conditions of intra_cluster.
% n2, propagation conditions of inter_cluster.
Maxihop=ceil(R/D)-1;
k=8.429*10^(-16);
e=50*10^(-12);

Ratio=1;
BitEnergy=(k*D^n1+1/Ratio*J*e+1/Ratio*J*k*3^n2*D^n2)*(D^2/R^2*(2*J-1));


function BitEnergy=BitEnergyFunction2(R,D,J,n1,n2)
% This function calculate bit energy cost given cluster size, number of
hops, and propagation
% R, The radius of the whole detection circle(field).
% D, cluster size;
% J, number of hops from the origin;
% n1, propagation conditions of intra_cluster.
% n2, propagation conditions of inter_cluster.
Maxihop=ceil(R/D)-1;
k=8.429*10^(-16);
e=50*10^(-12);
Ratio=1;
BitEnergy=(k*D^n1+1/Ratio*J*e+1/Ratio*J*k*3^n2*D^n2)*((R^2-
(Maxihop*D)^2)/R^2);
```